Optimal Structural Design of a Planar Parallel Platform for Machining

by

Craig Stephen Long

A thesis submitted in partial fulfillment of the requirements for the degree

Master of Engineering

*in the Faculty of Engineering, the
Built Environment and Information Technology
of the
University of Pretoria*

November 2001

Abstract

Title: Optimal Structural Design of a Planar Parallel Platform for Machining

Author: Craig Stephen Long

Supervisor: Prof. J.A. Snyman

Co-Supervisor: Prof. A.A. Groenwold

Department: Department of Mechanical Engineering

Degree: Master of Engineering

Keywords: Stewart platform, planar parallel manipulator, optimization, genetic algorithm (GA), LFOPC, Dynamic-Q, finite element method (FEM), stiffness mapping

Parallel manipulators have many advantages over traditional serial manipulators. These advantages include high accuracy, high stiffness and high load-to-weight ratio, which make parallel manipulators ideal for machining operations where a high accuracy is required to meet the requirements that modern standards demand.

A number of previous workers have determined the stiffness of parallel platforms using the duality between instantaneous kinematics and statics in parallel mechanisms. For the aforementioned analysis, compliance is introduced in the actuators, resulting in a platform stiffness matrix. This methods furthermore predicts when the platform approaches a singular or ill-conditioned configuration.

However, this idealized estimate of the stiffness is not accurate enough to determine how an actual platform assembly will react to an externally applied force. For a planar parallel platform, the out of plane stiffness is not included in the resulting stiffness matrix since the kinematics equations are derived only in the plane in which the platform operates.

Recently, the finite element method (FEM) has been used by some workers to determine the stiffness of spatial manipulators. These models are mainly used to verify stiffness predicted using kinematic equations, and are restricted to relatively simple truss-like models. In this study, state-of-the-art finite elements are used to determine the out of plane stiffness for

parallel manipulators. Beam elements that make use of Timoshenko beam theory and flat shell elements with drilling degrees of freedom are used to model the platform assembly.

The main objective of this study is to quantify the stiffness, particularly the out of plane stiffness, of a planar parallel platform to be used for machining operations. The aim is to suggest a design that is able to carry out machining operations to an accuracy of 10 μm for a given tool force.

Reducing the weight of a parallel manipulator used in machining applications has many advantages, e.g. increased maneuverability, resulting in faster material removal rates. Therefore the resulting proposed design is optimized with respect to weight, subject to displacement and stress constraints to ensure feasible stiffness and structural integrity.

This optimization is carried out with both gradient-based methods and a genetic algorithm (GA). The gradient-based methods include LFOPC and Dynamic-Q. A binary GA, implemented as both a micro GA and full GA, is used to provide for the future inclusion of discrete design variables.

Stiffness maps, as proposed by Gosselin, are drawn for the optimal design. These stiffness maps can aid in determining the best toolpath inside a feasible workspace.

It is envisaged that this work, together with current work at the University of Pretoria, will result in a feasible design for a planar parallel platform to be used in industry. An application of such a planar parallel platform lies in retro-fitting existing, relatively inexpensive 3-axis milling equipment. This increases their capability at a lower cost than the of the alternative of purchasing a traditional 5-axis milling center.

Opsomming

Titel:	Optimale Strukturele Ontwerp van 'n Vlak Parallelle Platform vir Masjinering
Outeur:	Craig Stephen Long
Leier:	Prof. J.A. Snyman
Mede-leier:	Prof. A.A. Groenwold
Departement:	Departement van Meganiese Ingenieurswese
Graad:	Meester van Ingenieurswese
Sleutelwoorde:	Stewart platform, vlak parallelle manipuleerder, optimering, genetiese algoritme (GA), LFOPC, Dynamic-Q, eindige element metode (EEM), styfheidstransformasie

Parallelle manipuleerders het talle voordele bo tradisionele serie manipuleerders. Die voordele sluit hoë akkuraatheid, hoë styfheid en 'n hoë las-tot-gewigverhouding in, wat parallelle manipuleerders aantreklik maak vir masjineringsaksies waar 'n hoë akkuraatheid vereis word om aan die vereistes van moderne standaarde te voldoen.

'n Aantal werkers het voorheen die styfheid van parallelle platforms bepaal deur gebruik te maak van die dualiteit tussen oombliklike kinematika en statika in parallelle meganismes. Vir eersgenoemde analise word die styfheids-inverse van die aktueerders bereken, wat 'n resultante styfheidsmatriks gee. Hierdie metode voorspel voorts ook wanneer die platform singulêre of swak-gedefiniëerde konfigurasies nader.

Die geïdealiseerde benadering van die styfheid is egter nie akkuraat genoeg om te bepaal hoe 'n werklike platform samestelling reageer onder eksterne belastings nie. Vir 'n vlak parallelle platform word die uit-vlak styfheid nie in die resulterende styfheidsmatriks ingesluit nie, aangesien die kinematiese verwantskappe slegs afgelei word in die vlak waarin die platform opereer.

Die eindige element metode (EEM) is onlangs deur sommige werkers gebruik om die styfheid van ruimtelike aktueerders te bepaal. Hierdie modelle word hoofsaaklik gebruik om styfheids voorspellings gedoen met behulp van inverse kinematiese verwantskappe te kontroleer, en is beperk tot relatief eenvoudige stang-agtige modelle. In hierdie studie word gevorderde eindige elemente gebruik om die uit-vlak styfheid van parallelle manipuleerders te bepaal. Balk elemente wat op Timoshenko balk teorie gegrond is, en plat dop elemente met boorrotasie vryhede, word gebruik om die platform samestelling te modelleer.

Die hoofdoel van hierdie studie is die kwantifisering van styfheid, en in die besonder die uitvlak styfheid, van 'n parallelle platform vir masjinerings doeleindes. Daar word gepoog om 'n ontwerp geskik vir 'n masjinerings proses binne 'n toleransie van 10 μm vir 'n voorgeskrewe gereedskapslas daar te stel.

Die verlaging van die gewig van 'n parallelle manipuleerder wat in masjinerings prosesse gebruik word het talle voordele, soos bv. verhoogde manufreerbaarheid, wat tot hoër tempos van materiaalverwydering lei. Die resulterende voorgestelde ontwerp word dan ook geoptimeer met betrekking tot gewig, onderhewig aan verplasings- en spanningsbegrensings wat voldoende styfheid en strukturele integriteit waarborg.

Hierdie optimerings proses word met beide gradiënt metodes en 'n genetiese algoritme (GA) uitgevoer. Die gradiënt metodes sluit LFOPC en Dynamic-Q in. 'n Binêre GA wat as beide 'n mikro en 'n vol GA geimplementeer is, word gebruik om voorsiening te maak vir die insluiting van diskrete ontwerpsveranderlikes in die toekoms.

Styfheidskaarte, soos voorgestel deur Gosselin, word opgestel vir die optimale ontwerp. Hierdie styfheidskaarte kan behulpsaam wees wanneer die beste gereedskapspad binne 'n wettige werksruimte gevind moet word.

Dit word verwag dat hierdie werk, tesame met ander projekte by die Universiteit van Pretoria, sal lei tot 'n werkbare vlak parallelle platform vir gebruik in die industrie. 'n Toepassing van so 'n vlak parallelle platform lê in die ombouing van bestaande, relatief goedkoop 3-assige freesmasjiene. Dit verhoog die vermoëns van hierdie masjiene teen 'n laer koste as die alternatief, naamlik die aanskaf van 'n tradisionele 5-assige masjien.

Acknowledgments

I would like to express my sincere gratitude towards the following persons:

- My Supervisor, Professor Jan Snyman, for his invaluable guidance and support during the course of this study.
- My Co-Supervisor, Professor Albert Groenwold, for his many valued inputs.

Financial support granted by the National Research Foundation (NRF) is gratefully acknowledged.

- To Lukas Du Plessis for his insight and willingness to help.
- The contributions of Alex Hay and Willem Smit are also much appreciated.
- To my parents who have given me every opportunity possible - Thank you.
- Sincere thanks to Michele without whom this work would not have been possible. I hope one day to repay your selfless generosity.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The word “robot” is, to a large extent, associated with the robots seen in movies. However, in practice most robots are used as manufacturing tools, performing tasks such as spray-painting, welding, assembling and machining.

In general, robots may be divided into two categories, namely *serial* and *parallel* manipulators. Serial manipulators are mechanisms in which a so-called end-effector is connected to ground by only one kinematic chain. The end-effector may be defined as a tool or gripping mechanism attached to the wrist of a robot to accomplish some prescribed task. It usually encompasses a motor, or driven mechanical device. It may be used as a sensor, gripping device, paint gun, drill, welding device, etc. [1]. Figure 1.1 depicts a typical serial manipulator. (Figure 2.2(b) in Chapter 2 depicts a parallel type manipulator.)

Of particular importance in the study of serial and parallel manipulators is their *dexterity* and associated *workspace*. Dexterity is a measure of a robot's ability to follow complex paths, while workspace may be defined as the volume (or area in the case of planar manipulators) of space within which the robot can perform given tasks [1, 2].

Serial-link manipulators are used in applications where a large workspace is required with increased dexterity therein. However, their cantilever structure makes them susceptible to large deflections, resulting in poor stiffness and thus poor positioning accuracy.

In recent years, parallel manipulators, and in particular Stewart platforms, have been receiving a great deal of attention in the field of robotics research. An application particularly suited for this type of manipulator lies in the material machining industry, since parallel manipulators do not suffer from the aforementioned stiffness drawbacks suffered by serial manipulators.

Parallel manipulators are used in applications in which the accuracy requirements outweigh the need for a large workspace [3]. For material removal machine tools, higher stiffness allows higher machining speeds and feed rates while maintaining the desired precision, surface finish, and improved tool life.

The most useful advantage of parallel mechanisms over serial type manipulators is their

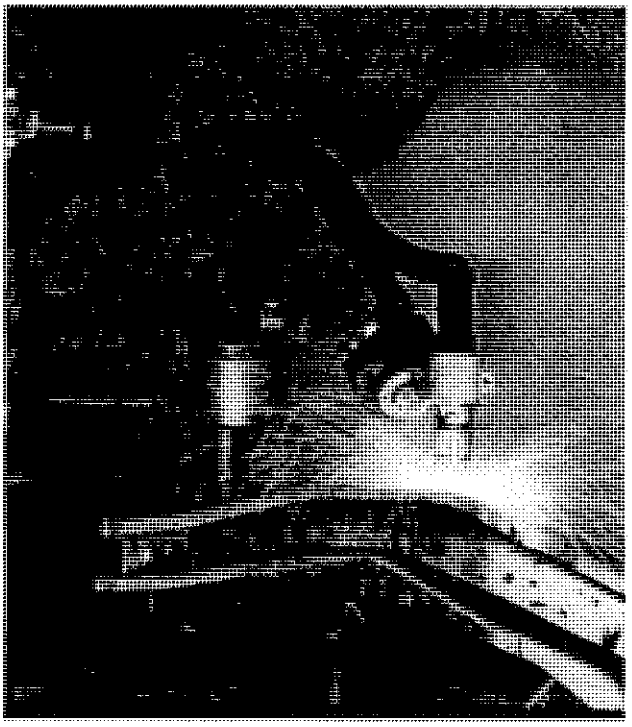


Figure 1.1: Typical serial manipulator.

ability to carry a high load, while not becoming bulky and heavy. This is due to the fact that loads are distributed, to a great extent, evenly through the actuators. A further advantage is the fact that the actuators experience mostly traction and compression loads, which are preferred when using linear type actuators.

Another advantage of a parallel manipulator over a serial type is the fact that the end-effector is much less sensitive to the errors in the position of the actuators. In a parallel type manipulator, the error is approximately equal to the average of the actuator errors. In a serial type actuator, the error of the end-effector is equal to the sum of the errors of the actuators. This makes a parallel-link manipulator scheme, for a machining tool, potentially more accurate and repeatable. Also, the superior robotic stiffness of the structure results in the deformation of the links being minimal, and contributes to the high positional accuracy of the end-effector.

Thus, since in-parallel actuated mechanisms have advantages in accuracy, operational speed, payload, rigidity and output power, compared with serial mechanisms, parallel manipulators are well suited to heavy duty applications [4].

1.2 Objectives

This study forms part of a larger project at the University of Pretoria, in which the feasibility of retro-fitting an existing 3-axis milling machine with a planar parallel platform is being investigated. This will effectively transform a relatively inexpensive 3-axis milling machine into a 4 axis machine center which is capable of machining more complex and irregular shapes. Adding a fifth (rotation) axis to such a machine can be accomplished relatively inexpensively, resulting in a machine capable of manufacturing complex shapes, e.g. plastic injection molds, thus adding greatly to the value of the machine.

Researchers at the University of Pretoria [5, 6, 7] have studied various aspects of this type of manipulator, including workspace and singularity considerations. In Section 2.2.8 this existing work is presented in more detail.

Merlet [8] points out that the design process entails determining the design parameters so that a set of constraints is satisfied. He lists several possible constraints for the design of a manipulator, including:

- a workspace requirement,
- the maximum accuracy over the workspace for a given sensor accuracy,
- the minimum articular forces for a given load set,
- the maximal stiffness of the robot in some direction, and
- the maximum velocities or accelerations for given actuator velocities and accelerations.

The current work deals with the design of a low weight planar parallel platform, subject to stiffness and stress constraints. An existing design is to be critically analyzed in order to determine whether a machining tolerance of 10 μm is possible. If this tolerance is not met, a change in design is to be suggested which *is* capable of meeting this tolerance. Once the design is capable of meeting the prescribed tolerance, it is to be optimized with respect to weight. The requirements may thus be summarized as:

1. To analyze and critically evaluate an existing planar parallel platform design.
2. To suggest, if necessary, design improvements such that the prescribed machining tolerance of 10 μm is attainable for a given tool force.
3. To minimize the weight of the manipulator, subject to stiffness and stress constraints, and such that the prescribed machining tolerance is met.

1.3 Approach

Traditionally, the stiffness of a Stewart platform has been computed using the kinematic Jacobian matrix [9]. Recently the finite element method has also been used to perform the stiffness analysis [10]. These fairly simple finite element analyses, using mainly simple truss elements, yield results of similar accuracy to that of the Jacobian stiffness analysis, and are used mainly to verify stiffness predicted using kinematic equations.

For the application studied here, the Jacobian stiffness analysis is not sufficient, since it cannot calculate the out of plane displacements of the *planar* platform. The development of flat shell elements with drilling degrees of freedom [11, 12, 13], allowing beam-plate connections, gives designers the ability to accurately model a planar parallel platform. Furthermore, using beam elements instead of simple truss elements enables the calculation of out of plane deflections accurately.

For these reasons, a more complex model of the planar platform is used in this study to calculate the platform displacement and stiffness. Optimization with respect to weight is then performed using both a Genetic Algorithm (GA) [14] and gradient-based methods [15, 16]. The results of the different optimization algorithms are analyzed and compared. The GA is included to facilitate the envisaged future optimization using discrete variables, since in the discrete case the design space is disjointed and gradient information is of little use.

The optimization results are furthermore used to evaluate the existing planar platform design and to suggest design improvements subject to a machining tolerance of 10 μm being maintained.

1.4 Thesis overview

In Chapter 2 a brief overview of existing work regarding parallel platforms, and the Stewart platform in particular, is presented. Firstly, a brief history of the Stewart platform is presented and typical applications for this manipulator are discussed. Thereafter the current research trends in this field of study are touched upon.

Chapter 3 starts with a description of the existing planar parallel platform design, as well as some detail concerning the way in which the platform is modeled. The chapter begins with a description of the individual components of the planar platform assembly, and describes how each is modeled. Next the model boundary conditions are presented, followed by some results from the preliminary finite element analyses.

Chapter 4 is concerned with the optimization of the planar parallel platform proposed in Chapter 3. The formal formulation of the optimization problem is presented, followed by a discussion of the optimization methodology that is to be used. The chapter concludes with a presentation of the optimization results, followed by a discussion of the results and the conclusions drawn. It is ultimately shown that the design is unable to meet the prescribed accuracy.

Chapter 5 contains information regarding the proposed improved planar parallel platform design to meet the prescribed accuracy. Firstly a description of the improved design is presented followed by a summary of the finite element model that is used. Next, the optimization results for this design are presented and discussed. Finally, stiffness maps are calculated for the improved design and the importance of toolpath planning is touched on.

Finally the study's overall conclusions and proposals for future work are presented in Chapter 6.

Appendix A presents the traditional approach to computing the robotic stiffness (to be referred to as the Jacobian method) required for the static analysis of a parallel platform, with compliant actuators.

Appendices B and C present more detail regarding the respective formulation of the frame and flat shell elements used in the finite element analyses.

Appendices D, E and F detail the optimization algorithms that are used in this work. First the two gradient-based methods originated by Snyman, namely the "Leap-Frog" and Dynamic-Q methods are presented. This is followed by a presentation of the Genetic Algorithm as coded by Carrol.

Appendix G contains information regarding the calculation of the tool forces that are assumed to be acting on the workpiece.

Appendix H depicts the sensitivity of the objective and constraint functions to the various variables.

Chapter 2

Overview of Existing Work

In this chapter existing work regarding Stewart platform manipulators is briefly summarized. Current research trends and existing work at the University of Pretoria are noted and briefly discussed.

2.1 Brief overview of Stewart platform manipulators

This section contains a brief introduction to Stewart platforms and contains information regarding the history, uses and some of the various designs of parallel manipulators.

2.1.1 History of the Stewart platform

In 1965, Stewart [17] suggested a parallel manipulator with six degrees of freedom (now commonly known as a Stewart platform). The original application suggested in this pioneering paper was to use the manipulator as a flight simulator.

One of the responses to the paper of Stewart was from Gough who stated that the design of Stewart's was similar to a tire testing machine that Gough and Whitehall [18, 19] had already designed and built by 1955. This type of parallel manipulator is therefore also often referred to as a Gough-Stewart platform.

Though the Stewart platform was proposed by Stewart [17] in 1965, its popularity as a research topic only developed in the 1980's. In the last decade there has been a steady increase in the interest shown in parallel manipulators, and especially Stewart platform type manipulators, for a number of different applications [20].

2.1.2 Introduction to parallel manipulators

Since traditional serial manipulators are, in general, more commonly used in industry, the use of parallel manipulators in their place needs to be adequately motivated. In this section, some of the salient features of parallel manipulators are pointed out. Additionally, several of the parallel manipulators which have already been designed and built are listed.

Merlet [21] defines a parallel manipulator as "*a closed-loop mechanism in which the end-effector is connected to the base by at least two independent kinematic chains*". He defines a fully-parallel manipulator as "*a closed-loop mechanism with an n degree-of-freedom end-effector connected to the base by n independent chains which have at most two links and are actuated by a unique prismatic or rotary actuator*". A Stewart platform is a fully-parallel manipulator with 6 degrees of freedom (dof).

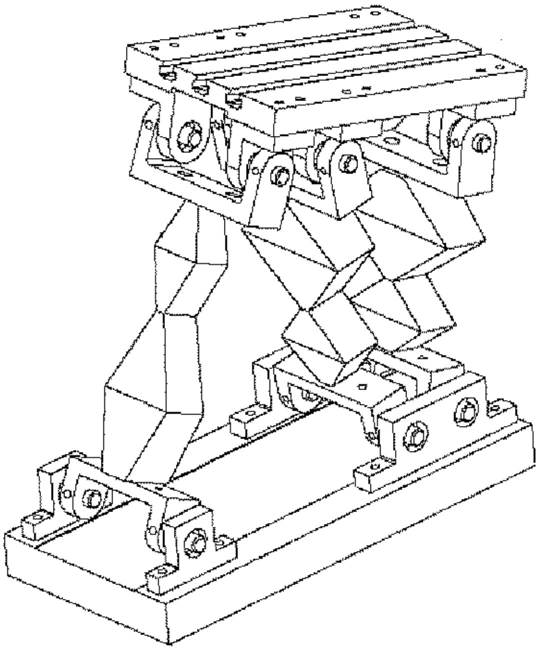


Figure 2.1: SMARTCUTS 3-dof planar parallel platform.

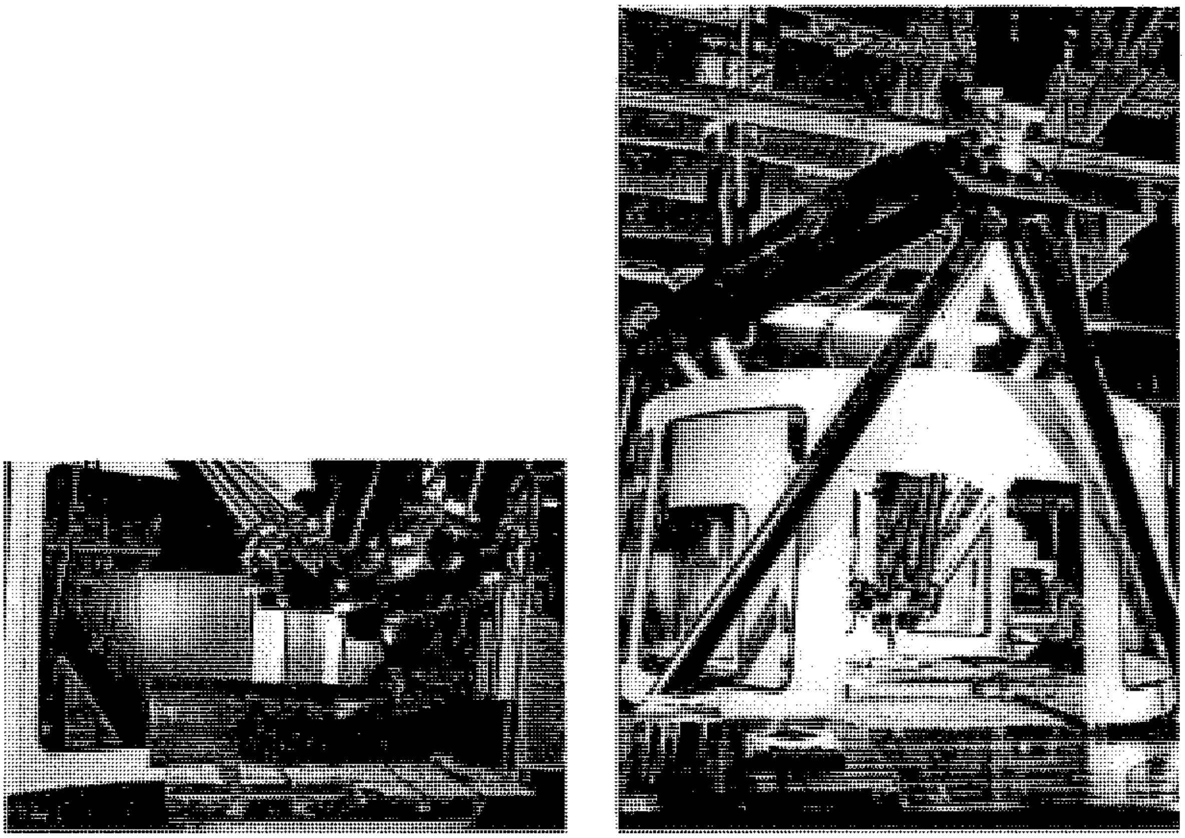
Although it is relatively simple to design a serial manipulator to carry out complicated tasks, these types of manipulators are susceptible to large deflections and vibration problems at high speeds due to their low rigidity. Parallel manipulators do not suffer from these disadvantages.

Since the load is distributed relatively evenly between its actuators and furthermore, the actuators experience mostly traction and compressive loads, parallel manipulators have favorable load to weight ratios. As an example, Merlet [22] cites a parallel manipulator which he developed with a weight of 35 kg and a nominal load capacity of 600 kg.

Additionally, parallel manipulators are less susceptible to end-effector error due to actuator positioning errors. This is due to the fact that the end-effector error is approximately equal to the average of the individual actuator errors, unlike serial manipulators for which end-effector error is equal to the sum of the errors of the actuators. Parallel schemes, unlike serial manipulators with a cantilever-type construction, have high global stiffness. This high global stiffness contributes to the platform end-effector's high positional accuracy.

Thus, since in-parallel actuated mechanisms have advantages in accuracy, operational speed, payload and rigidity, compared to that for serial mechanisms, parallel manipulators are better suited to heavy duty applications [4].

A number of experimental parallel manipulators have been proposed and built, for example the SMARTCUTS three degree of freedom (3-dof) planar platform shown in Figure 2.1 and the 6-dof manipulators shown in Figures 2.2(a) and 2.2(b) [23]. Perhaps the most notable Stewart platform-based machine tool in existence is the *Octahederal Hexapod* machine tool developed by the *Ingersoll Milling Machine Company.* This milling machine consists of a moving tool assembly manipulated over the workpiece by six ball screw actuators, making up the 6-6 Stewart platform shown in Figure 2.2(b). The standard notation describing the number of connection points for a Stewart platform $(\alpha - \beta)$ is employed here. α refers to the number of moving platform connection points while β refers to the number of stationary platform connection points.



(a) Octahedral Hexapod machining wax rendition of composite panel forming die. (Part geometry supplied by NASA Johnson Space Flight Center).

(b) Octahedral Hexapod machine installed at the National Institute of Standards and Technology (NIST).

Figure 2.2: Examples of 6-dof Stewart platform based machine tools.

The internet web site "PKM (Parallel Kinematic Machines) Bookmarks" [24] lists a number of research organizations involved in the development of parallel platforms, as well as parallel platforms either available or in development in industry. Some of the research organizations and their available products are listed below.

Products developed at research organizations:

- NASA (*Cockpit Motion Facility*)
- LIRMM (*Parallel Robots*)
- LME (*Hexapod Machine*)
- INRIA (*Left hand, Active wrist*)
- Sandia Hexapod Testbed
- NIST (*Hexapod*)
- NIST (*RoboCrane*)
- SFIT (*Hexaglide*)

Products developed in industry:

- Toyoda (*HexaM Machine*)
- DEMAUREX, Swiss (*Delta, flexible packaging systems*)
- Hexel Corporation (*Tornado 2000*)
- AEA Technology (*Joystick*)
- Vertex Antennentechnik (*Antennas, Radio telescopes*)
- CAE Electronics Ltd. (*Motion simulators*)
- MOOG Motion Systems (*Motion simulators*)
- ViRtogo (*Motion Simulators*)
- DELTALAB (*Educational Set EX800*)

In addition to these organizations, a number of learned institutions and universities are also involved in parallel platform development. This suggest how popular and important this line of research has become. Yet, in spite of this activity Ji [25] reports as recently as 1996, that the use of platform manipulators is still mainly in an experimental stage.

2.1.3 Uses of the Stewart platform

Parallel-link manipulators based on the Stewart platform have been proposed as schemes for many different applications, including:

- machine tools,
- flight simulators and amusement park rides,
- assembly applications which require large forces,
- forming presses,
- mobile robots (by having a variable base geometry) and
- and for use in stabilizing platforms aboard ships.

2.2 Current research trends

In their state-of-the-art reviews, Merlet [21], and more recently, Dasgupta and Mruthyunjaya [26] identify a number of current research trends with regard to the Stewart platform and other parallel platforms. In this section these trends are briefly discussed in order to put the present work into perspective.

2.2.1 Direct position kinematics

The forward kinematics problem may be simply stated as: *given the leg or link lengths, determine the end-effector or platform position and orientation.* This problem is challenging since the solution is not unique and an iterative approach is required in most cases.

The mathematical description of the problem equates to the solution of the kinematic equations [27]:

$$\sqrt{(\bar{b}_i - (\bar{d} + \boldsymbol{R}\bar{p}_i^m))^T(\bar{b}_i - (\bar{d} + \boldsymbol{R}\bar{p}_i^m))} = l_i, \qquad i = 1, 2, \ldots, 6 \tag{2.1}$$

where:

$\bar{b}_i$ represents the vector from the origin of the base coordinate frame $\boldsymbol{B}$, to the joint that connects the i^{th} leg to the base,

$\bar{p}_i^m$ is the vector that connects the coordinate system of the platform $\boldsymbol{M}$ with the i^{th} leg joint in the platform coordinate system,

$\bar{d} = [x, y, z]$ is the position vector of the platform coordinate system's origin ($\boldsymbol{M}$) in the base coordinate system $\boldsymbol{B}$,

l_i is the length of the i^{th} leg, and

$$\boldsymbol{R} = \begin{bmatrix} \cos\phi\cos\theta & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi & \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi \\ \sin\phi\cos\theta & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi \\ -\sin\phi & \cos\theta\sin\psi & \cos\theta\cos\psi \end{bmatrix}$$

is the rotation matrix relating the platform's coordinate system, $\boldsymbol{M}$, to the base coordinate system, $\boldsymbol{B}$ as shown in Figure 2.3. Here $\boldsymbol{R}$ is constructed using Roll (ϕ) about the z-axis, Pitch (θ) about the y-axis and Yaw (ψ) about the x-axis. In particular, it is required to solve for $\bar{d} = [x, y, z]$ and the angles ϕ, θ and ψ.

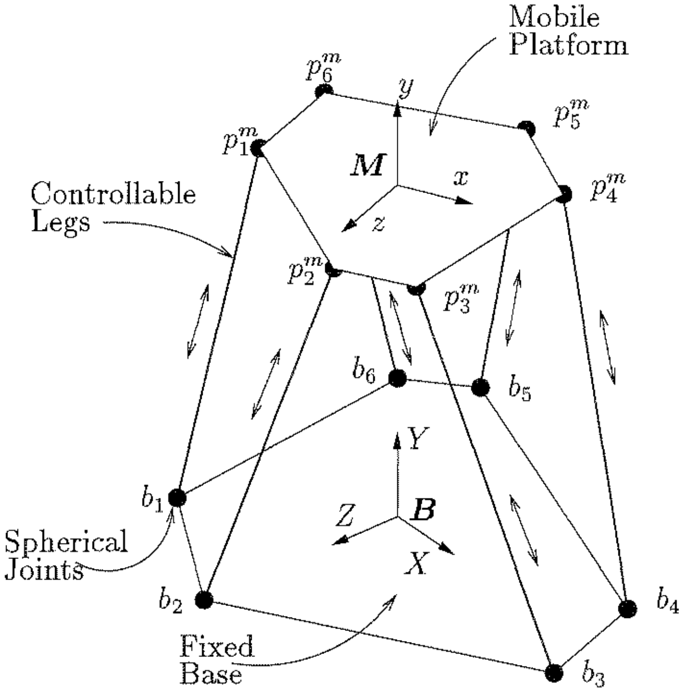


Figure 2.3: Schematic of a 6-dof Spherical Prismatic Spherical (SPS) Stewart platform.

Merlet [28] developed an algorithm to solve the Stewart Platform forward kinematics problem, in which the upper bound on the number of solutions was found to be 1320. Raghavan [29] found that a general Stewart platform has 40 real solutions. Some special designs of the Stewart Platform reduces the solution space. Utilizing specialized approaches, many different structures, ranging from the simplest (3-3) to complicated (5-5 and 6-4) schemes, were solved. In [30, 31, 32] and many others, authors studied specific topological configurations of the Stewart Platform. As El-Khasawneh and Ferreira [27] point out, these solutions are highly restrictive or even mechanically infeasible designs.

Gosselin *et al.* [33] studied a 3-dof planar manipulator, in which he found that reducing the degrees of freedom to 3 greatly simplifies the problem. The forward kinematics problem for this manipulator becomes a matter of solving a 6^{th} order polynomial in one variable. Simplifying the design further, by having the three base joints collinear and the platform joints collinear reduces the order of the polynomial to 3^{rd} order, which has a closed-form solution. They explain that having a closed-form solution for the forward kinematics problem will impact the control type and sampling speed that can be implemented. This has implications for the accuracy and dynamic stiffness that can be achieved through closed loop controls. This implies that lower dof parallel link manipulators may have superior performance to a full 6-dof manipulator in some respects.

In summary, closed-form solutions of special cases, numerical solution schemes and analytical approaches have been implemented to solve the direct positional kinematics problem.

2.2.2 Statics and rate kinematics

The inverse kinematics problem is defined as: *given the end-effector position and orientation, compute the manipulator leg-lengths.* Unlike the forward kinematics problem, the inverse kinematics problem is easy to solve [34]. This problem is defined mathematically as:

$$l_i = \sqrt{(\bar{b}_i - (\bar{d} + \boldsymbol{R}p_i^{\bar{m}}))^T(\bar{b}_i - (\bar{d} + \boldsymbol{R}p_i^{\bar{m}}))} = g_i(\boldsymbol{R}, \bar{d}) \tag{2.2}$$

where l_i, $\bar{b}_i$, $\bar{d}$, $p_i^{\bar{m}}$, and $\boldsymbol{R}$ are as described in (2.1).

Now, the change in the length of the i^{th} leg, Δl_i is obtained as a row vector $\boldsymbol{J}_i$ multiplied by a column displacement vector Δp as shown below:

$$\Delta l_i = \boldsymbol{J}_i \Delta \boldsymbol{p} \tag{2.3}$$

where $\boldsymbol{J}_i = \begin{bmatrix} \frac{\delta g_i}{\delta x} & \frac{\delta g_i}{\delta y} & \frac{\delta g_i}{\delta z} & \frac{\delta g_i}{\delta \psi} & \frac{\delta g_i}{\delta \theta} & \frac{\delta g_i}{\delta \phi} \end{bmatrix}$ and $\Delta \boldsymbol{p} = [\Delta x \;\; \Delta y \;\; \Delta z \;\; \Delta\psi \;\; \Delta\theta \;\; \Delta\phi]^T$. Assembling the equations for all the legs in the mechauism gives,

$$\Delta \boldsymbol{q} = \boldsymbol{J} \Delta \boldsymbol{p} \tag{2.4}$$

where $\Delta \boldsymbol{q} = [\Delta l_1 \;\; \Delta l_2 \;\; \Delta l_3 \;\; \Delta l_4 \;\; \Delta l_5 \;\; \Delta l_6]$. In the limit, this equation relates the speed of the joints of the mechanism to the speed at the end-effector. Thus

$$\dot{\boldsymbol{q}} = \boldsymbol{J} \dot{\boldsymbol{p}} \tag{2.5}$$

Where $\boldsymbol{q} = [l_1 \;\; l_2 \;\; l_3 \;\; l_4 \;\; l_5 \;\; l_6]$ and $\boldsymbol{p} = [x \;\; y \;\; z \;\; \psi \;\; \theta \;\; \phi]$. Thus $\boldsymbol{J}_i$ is the i^{th} row of the Jacobian matrix $\boldsymbol{J}$, and represents the change in the i^{th} joint for a unit component displacement vector change.

From the principle of duality between force-torque and velocity fields, or what is commonly known as contragedience (see Section 2.2.9):

$$\bar{\tau} = \mathrm{J}^T \bar{f} \tag{2.6}$$

where $\bar{\tau} = [F_x \;\; F_y \;\; F_z \;\; Mx \;\; My \;\; Mz]^T$ is the end-effector wrench, the vector of forces experienced by the legs is $\bar{f} = [f_1 \;\; f_2 \;\; f_3 \;\; f_4 \;\; f_5 \;\; f_6]$, and J^T is the transpose of the Jacobian matrix.

Thus the forward force transformation of the Stewart Platform is a straightforward linear mapping as described by Merlet [28] and Fichter [34], and denoted by the matrix $\boldsymbol{H}$ in the following discussion. The columns of the 6 × 6 force transformation matrix essentially consist of the Plücker coordinates of the six legs. Naturally the inverse force decomposition requires the inverse of the matrix $\boldsymbol{H}$, giving rise to the possibility of static singularities [26].

The inverse velocity kinematics, derived by Fichter [34], is essentially a linear transformation given by $\boldsymbol{H}^T$. The forward velocity kinematics can be obtained by inverting $\boldsymbol{H}^T$. The Jacobian in the conventional sense is thus given by $\boldsymbol{H}^{-T}$ for the Stewart Platform. The acceleration kinematics can also be derived from this transformation [26].

2.2.3 Singularities

Gosselin [9] suggests that the classification of singularities, which can occur only in mechanical systems containing closed loop kinematic chains, correspond to a configuration for which there exists a certain set of nonzero generalized velocity vectors, or twists, of the gripper link that produce a velocity of zero at all of the actuators. Therefore the motion of the gripper link cannot be controlled by the motion of the actuators, and the stiffness of the manipulator in the direction corresponding to the nullspace of the Jacobian matrix will be zero.

The Stewart Platform experiences force singularities when the force transformation matrix $\boldsymbol{H}$ is rank deficient. When the platform is in such a position, the manipulator loses some degree(s) of constraint and becomes uncontrollable.

The important association of the conditioning of the static transformation with the stiffness of the Stewart Platform was studied through stiffness mapping by Gosselin [9] who also discusses the near-singular behavior (or ill-conditioning) leading to loss of stiffness.

A more general problem, posed in a global sense, is that of singularity avoidance in path planning between two end poses. This question has been addressed by Dasgupta and Mruthyunjaya [35] who formulated the singularity-free path planning problem for the Stewart platform and developed a strategy for planning well-conditioned paths in the workspace of the manipulator. This problem was also addressed by Snyman *et al.* [6] at the University of Pretoria and their work is discussed in Section 2.2.8. However rigorous criteria for the existence (or non-existence) of such a path is still not available.

2.2.4 Sensor applications

As the planar parallel platform structure has a good stiffness and the reconstruction of the wrench applied at the platform from measured leg forces is relatively straightforward, a Stewart platform with instrumented elastic legs can be used as a wrist force sensor.

Dasgupta *et al.* [36] presented a design methodology for the Stewart platform sensor structure based on the optimal conditioning of the force transformation matrix.

2.2.5 Workspace and dexterity

The relatively small workspaces of parallel manipulators is one of their major drawbacks. As such, the problem of solving for the workspace of these types of manipulators is not only important, but also a challenging problem.

The problem [26] is to "*determine the kinematic geometry of a Stewart platform manipulator for a singularity-free workspace segment with the given boundary and with orienting capabilities in a given 3D region*".

In short the following three possibilities may arise and need to be addressed:

- Given an orientation, or number of orientations in 3D space, the positional workspace of the end-effector is to be determined.
- Given a position, or a number of positions in 3D space, the orientating capability at these positions is to be determined.
- Given a trajectory, it is necessary to determine whether the trajectory lies completely within the manipulator workspace.

In addition, such an analysis should preferably be performed in association with singularity analysis because a workspace segmented by singularity barriers will not be fully usable in practice.

One of the major difficulties in the workspace and dexterity analysis of the Stewart platform is the strong coupling between the position and orientation.

According to Dasgupta *et al.* [26], Merlet [37] has already made great inroads with regards this problem, and expects that this problem will be attracting much attention in the near future.

2.2.6 Dynamics and control

Compared to the vast amount of literature on the kinematic analysis of the Stewart platform, literature concerning both the dynamics and the control of this manipulator are relatively few. Control of the Stewart platform manipulator is almost an open field and work reported is not rigorous [26].

The dynamics problem has been solved for a simplified 3-dof planar parallel platform model, in which frictionless joints and negligible dynamic effects due to asymmetrical legs are assumed [38].

Generally, the choice of generalized coordinates in the derivation of the dynamic equations is critical to ensure manageable equations. Since the forward kinematics problem of the Stewart platform is much more complicated than its inverse kinematics problem, the Cartesian position and orientation of the mobile platform cannot easily be expressed in terms of the leg lengths, whereas given the platform coordinates $[x \quad y \quad z \quad \psi \quad \theta \quad \phi]$ with respect to the global reference frame, it is a simple task to obtain the leg lengths. Thus the dynamic equations are less difficult to obtain in the Cartesian space than the joint space.

If $\boldsymbol{p} = [x \quad y \quad z \quad \psi \quad \theta \quad \phi]^T$ is the vector with the coordinates of the platform and $\boldsymbol{q} = [l_1 \quad l_2 \quad l_3 \quad l_4 \quad l_5 \quad l_6]^T$ is the vector with the leg lengths, the Jacobian relating the associated velocities are given by

$$\dot{\boldsymbol{q}} = \boldsymbol{J}(\boldsymbol{p})\dot{\boldsymbol{p}} \tag{2.7}$$

with $\boldsymbol{p}$ representing the generalized coordinates. The dynamic equations can be derived by the Lagrangian approach and can be written as [38]:

$$\boldsymbol{M}(\boldsymbol{p})\ddot{\boldsymbol{p}} + \boldsymbol{C}(\boldsymbol{p}, \dot{\boldsymbol{p}}) = \boldsymbol{J}^T\boldsymbol{\tau} \tag{2.8}$$

where $\boldsymbol{M}$ is the mass matrix, $\boldsymbol{C}$ is the vector with the coriolis, centrifugal and gravity terms and $\boldsymbol{\tau}$ is the forces acting in the legs. The above equation holds for a manipulator that is not interacting with the environment. If the manipulator is interacting with the environment and is exerting a force, $\boldsymbol{F}$, in the task space the dynamic equations can be written as [38]:

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{F} = \boldsymbol{J}^T\boldsymbol{\tau} \tag{2.9}$$

2.2.7 Stiffness analysis

The stiffness of a parallel manipulator has a direct impact on its usefulness as a machine tool since stiffness is related to the accuracy with which tasks are carried out. Stiffness also has implications on the life of the tool with which machining operations are carried out. If the platform is sufficiently stiff, undesirable phenomena such as tool chatter are less likely to occur.

The traditional robotic stiffness matrix may be written as [27]:

$$\boldsymbol{K} = \boldsymbol{J}^T\boldsymbol{k}\boldsymbol{J} \tag{2.10}$$

where $\boldsymbol{k} = \text{diag}(k_1\ k_2\ \ldots k_6)$ is a matrix containing the stiffness of each actuator, modeled as linear springs. The stiffness matrix is a positive semi-definite symmetric matrix whose eigenvalues represent the coefficient of stiffness in the principal directions (which are given by the eigenvectors). These directions are in fact represented by twist vectors, i.e., generalized velocity vectors [9].

Moreover, the square root of the ratio of the smallest eigenvalue to the largest one gives the reciprocal of the condition number κ of the Jacobian matrix [39], which is a measure of the dexterity of the manipulator. This is written as:

$$\frac{1}{\kappa} = \sqrt{\frac{\lambda_{min}}{\lambda_{max}}} \tag{2.11}$$

where λ_{min} and λ_{max} are the smallest and the largest eigenvalues of the stiffness matrix respectively [9].

Gosselin [9] developed stiffness or conditioning maps for a Stewart platform for a specific direction of perturbation. These stiffness maps are used to reveal the existence of zones where the stiffness is not acceptable, or in other words, when the manipulator is close to a singular configuration.

In their papers, El-Khasawneh and Ferreira [27, 40] address the problem of finding the minimum and maximum stiffness and the directions in which they occur for a manipulator in a given posture. Furthermore, they show how the stiffness value in any direction can be computed from the stiffness in the eigenvector directions. Engineers are often interested in the response of the mechanism in the direction of perturbation, and they prove that the computed bounds (maximum and minimum) are tight for such a deflection. Stiffness, computed using the algebraic formulae they derived, were compared to those obtained from a finite element analysis to demonstrate the correctness of the formulation. In order to compare the results, a necessary assumption is that the platform is completely rigid.

Clinton *et al.* [10] presented the development of a mathematical model describing the stiffness of a Stewart platform-based milling machine. Matrix structural analysis was used to determine the stiffness matrix. Estimations of the system parameters are determined through experimental stiffness measurements. A computer simulation was used to demonstrate how the developed stiffness model suggests an optimization process for tool-path planning.

The above discussion deals with only a few of the many published works dealing with parallel manipulator stiffness (eg. [3, 4, 41, 42]).

In this study, research is conducted using the *real* stiffness (approximated by a numerical method) and not the traditional robotic stiffness (described in Appendix A). Out of plane stiffness of the planar parallel platform is considered, and the platform itself is not assumed to be rigid.

2.2.8 Existing work at the University of Pretoria

The current study forms part of a larger project at the University of Pretoria. As said, the aim of the project is to establish the feasibility of retro-fitting existing 3-axis milling machines with planar parallel platforms in order to improve the versatility of these relatively inexpensive machines. Various aspects of this design have been studied by several authors [5, 6, 7] and are summarized in this section.

A demonstration model of a planar parallel platform, shown in Figure 2.4, has been built by Du Plessis [43]. This demonstrator has the advantage of being adjustable, in that the placement of the revolute joints can be varied to suit workspace (or other) requirements. The current work needs to accommodate this adjustability.

Optimization approach to the determination of the boundaries of manipulator workspaces.

Snyman *et al.* [5] presented an optimization approach to the computation of the boundaries of a planar parallel platform. This numerical method consists of finding a suitable radiating point within the output coordinate space and then determining the points of intersection of a representative pencil of rays, emanating from the radiating point, with the boundary of the accessible set. This was done by application of a novel constrained optimization approach that has the considerable advantage that it may easily be automated. This method was illustrated by its application to two planar mechanisms, namely a planar Stewart platform

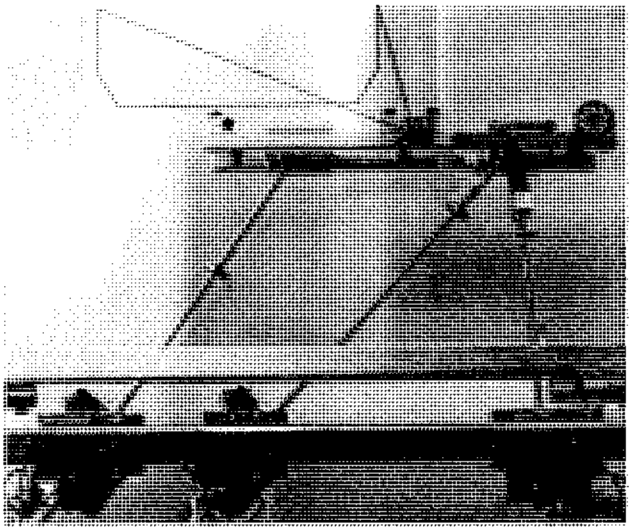


Figure 2.4: Planar parallel platform demonstrator at the University of Pretoria after completion of a specified toolpath.

and a planar redundant controlled serial manipulator. In addition the exterior boundaries of the workspace, interior curves that represent the configurations at which controlability and mobility may be limited, were also mapped. The optimization methodology, implemented for the planar case, may readily be extended to spacial Stewart platforms.

An extension to the novel optimization approach for the determination of accessible output sets of planar manipulators was presented by Hay *et al.* [7]. If any section of the workspace cannot be determined due to non-convexity, then the missing section is mapped using a suitably chosen new radiating point.

Optimization of the adjustable geometry of a planar Stewart platform machining center with respect to placement of workspace relative to toolpath.

Snyman *et al.* [6] presented a new integrated configurational design system that allows for the feasible and optimal placement of the workspace of an adjustable planar parallel manipulator, relative to a specified machine toolpath. The basic design of the manipulator is such that the positioning and relative size of the workspace may be varied by means of the adjustable placement of the revolute joints, shown in Figure 2.4. For a given specified toolpath an integrated geometry optimizer prescribes a manipulator configuration which not only ensures feasibility, but also enhances performance through the minimization of an objective function related to the required actuator forces and subject to physical constraints.

The availability of the combination of a physically adjustable platform coupled to a geometry optimizer, is essential for the practical application of a planar parallel platform as a machining device for toolpaths of varied specification. The proposed system was applied and tested through its application to a planar Stewart platform demonstrator with full adjustment capabilities. The platform serves to demonstrate the optimum machining motion obtainable through the integrated design system.

2.2.9 Series-parallel duality

As pointed out by Dasgupta and Mruthyuujaya [26], two celebrated theorems in mechanics, by Chasles aud Poinsot respectively, state that a general displacement of a rigid body can reduce to a twist about a screw and that a general force system can be reduced to a wrench on a screw. Upon these two theorems, Ball [44] developed the theory of screws.

Oue of the salient features of the theory of screws is the emergence of the duality between instantaneous kinematics and statics, angnlar and liuear velocities being dual to force and moment respectively.

It has been shown that there is a symmetry between serial chain and fully parallel systems such as the Stewart platform. This symmetry is showu to be the result of the duality of motion screw axes and wreuches. The appearance of the inverse of the Jacobian matrix in force decomposition, in the same role as the Jacobian in rate decomposition, is also a consequence of this same duality and of the reciprocity relationship between the motion of screw system, and the wrench system of a kinematic joint [45].

There are several symmetries between the force/torque and angular velocity/velocity vector systems. Classically it has been observed that the velocity field of a system of interconnected rigid bodies can be described by systems of instantaneous screw axes. At the same time the static force systems acting on the same system of rigid bodies can be described by a system of wrenches, which are vectorially homologous to the screw systems. The two types of vector systems are interrelated by Ball's reciprocity relationship [44, 45].

This duality between serial and parallel manipulators is used to summarize the positive and negative attributes of serial and parallel manipulators. Since most traditional machine tools are serial, the use of parallel manipulators in their place should be justified. El-Khasawneh [27] summarizes the following differences between parallel-link manipulators (PLM's) and serial-link manipulators (SLM).

Motion Description

The end-effector of a parallel-link manipulator moves at the intersection of the constraints produced by the joints. Serial-link manipulators end-effector motion is at the conjunction of the constraints produced by the joints.

Dynamics and Kinematics

The inverse dynamics and kinematics of SLMs are difficult to solve, while their forward problem is easy to solve. In coutrast, the inverse dynamics and kinematics problems of PLMs are easy to solve, but their forward dynamics and kinematics are difficult to solve, requiring the inversion of the Jacobian matrix.

Stiffness

SLMs have a cantilever-type structure, therefore their stiffness is low. The reason for this is that each actuator experiences the full end-effector load in addition to the weights of other joints and actuators. PLMs have a table-like structure, therefore, the stiffness is generally expected to be high. The end-effector load is distributed among the different legs, resulting in a smaller load per leg, producing smaller deflections, hence higher structural stiffness.

Singularity

Singular (degenerate) positions have different implications for these two classes of mechanisms. In SLMs, a singularity results in the loss of mobility in certain directions and partial locking. In PLMs it results in the gain of uncontrollable mobility in certain directions. This causes the stiffness to reach zero in those directions.

Accuracy

Accuracy is related to stiffness, the higher the stiffness the better the accuracy. Therefore the accuracy, in non-singular regions, is low for SLMs and high for PLMs. Furthermore, for serial-link manipulators the joint errors accumulate to the end-effector, often with large lever arm effects. In PLMs their effect is more direct and less accumulated and less magnified.

Workspace

Due to the SLM cantilever-type structure, and since the workspace size is dependent on the collective extendabilties of all the links, the end-effector reachability and workspace will be large. PLMs, on the other hand, have a table-like structure with limited workspace size that is dependent on the extendability of its individual legs.

Dexterity

SLMs have high dexterity which stems from good reachability and number of degrees of freedom. PLMs have low dexterity because of the small reach and mechanical constraints that are more limiting than they are in SLMs.

Torque and force variation

Since SLMs have a cantilever structure, joint torque requirements may vary substantially from joint to joint, depending on where in the serial order the joint is located. In PLMs the joint torque requirements are more uniform, because of the system symmetry. This allows for modularization of actuator units which may be an important manufacturing consideration.

Actuated and unactuated joints

In SLMs there are no unactuated joints. In PLMs it is necessary to have unactuated joints, such as spherical or revolute joints. Consequently the undesirable effects of clearances and backlash may be more pronounced in PLMs than SLMs.

Summary

The above discussion suggests that, while there are potential benefits for parallel-link manipulators, the size effects may limit the usefulness under certain circumstances. Given poor workspace sizes, it is possible that for some size machine the structural rigidity of the mechanism can be offset by the elasticity of the materials used in its construction. The effects of clearances, and the small workspace and possible poor conditioning of the workspace, suggest that the lower degrees of freedom PLMs may perform better. This further points to the possibility that serial-parallel hybrid schemes may provide a better blend of the positive features of both the SLMs and the PLMs.

Chapter 3

Modeling and analysis of an existing concept design

3.1 Introduction

Due to its unique configuration, the Stewart platform parallel manipulator distributes service loads (evenly) in the axial direction of the actuators. This property results in the platform's high global stiffness, and makes the Stewart platform manipulator very attractive for machining processes, where a high stiffness is required in order to meet the tight tolerances that modern standards demand. A possible application currently being explored is to retro-fit existing 3-axis milling machines with a planar parallel platform as shown in Figure 3.1.

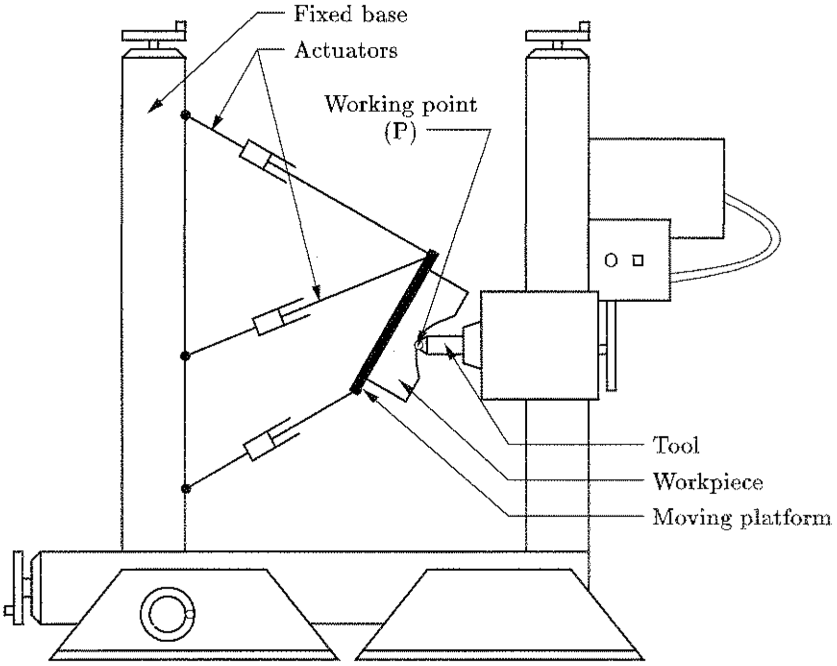


Figure 3.1: Schematic of milling operation using a planar parallel platform.

i 16031775
b15431204

In the case of the *planar* parallel platform, the high stiffness required is present only in the plane in which the moving platform operates, since the actuators act only in the plane of motion. This poses a problem if there are substantial forces experienced out of this plane of motion, such as those expected during milling operations. The stiffness in the out of plane direction is associated mainly with the bending resistance of the actuators.

The existing design, originated by Du Plessis [43], aimed at eliminating excessive out of plane displacements, is to add sliding guide rails adjacent to each of the three actuators. This proposed design is analyzed using FEM, and evaluated for machining purposes. If the design is found to be infeasible for machining operations, design improvements are required to meet the prescribed tolerance.

The FEM code EDSAP, originated by Wilson [46], is used to perform the finite element analyses. Any other finite element package (including commercial FEM packages) can be used in this process, however using EDSAP gives access to source code facilitating small element modifications if necessary. Furthermore, very accurate and thoroughly tested elements have been incorporated into the EDSAP suite of elements by Groenwold *et al.* [12, 13].

In order to explain the original concept design, a schematic figure of the assembly is depicted in Figures 3.2. For clarity, Figure 3.2 does not include the moving platform slab.

3.2 Components of the planar Stewart platform assembly

With reference to Figure 3.2, depicting the planar parallel platform assembly, the individual components are as annoted in Table 3.1. A discussion of the individual components of the assembly now follows.

Label	Component
A	Lateral platform members
B	Transverse platform members
C	Main transverse support struts
D	Main lateral support struts
E	Linear bearings
F	Guide rails
G	Actuator shafts
H	Actuator stepper motors
I	Transverse support strut bearings

Table 3.1: List of labels.

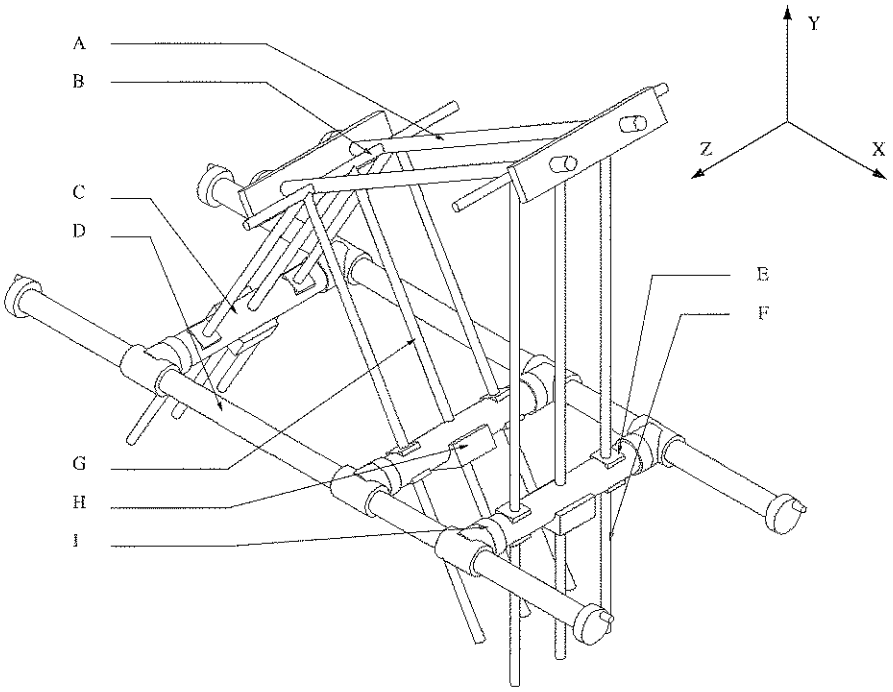


Figure 3.2: Originally proposed planar parallel platform design.

3.2.1 Actuators

The actuators (G & H) are of ball-screw type, similar to those shown in Figure 3.3 and are fixed to the transverse platform members at their upper ends.

The actuator shafts can not rotate as they are fixed to the transverse platform members which represent the upper revolute joint axes. The actuators are driven by stepper motors, in turn manipulating the moving platform. The driving nuts and motor assemblies are supported by the main transverse supporting struts which represent the lower revolute joint axes.

This type of actuator is capable of resisting and supplying high axial loads. However, tangential loads need to be avoided to prevent locking. The actuators are modeled as beam elements in order to accurately calculate the reaction forces and moments at the driving nuts. These forces and moments can later be used to determine if there is a possibility of the actuators locking, and whether or not fatigue may become a factor in future.

3.2.2 Guide rails

The linear bearing and rail (E & F) are required to supply sufficient resistance to out of plane bending, without interfering with the motion of the moving platform in the plane.

The actuators and sliding guide rails are attached to the same revolute joint axes above and below, therefore their respective effective lengths and orientations are necessarily equal.

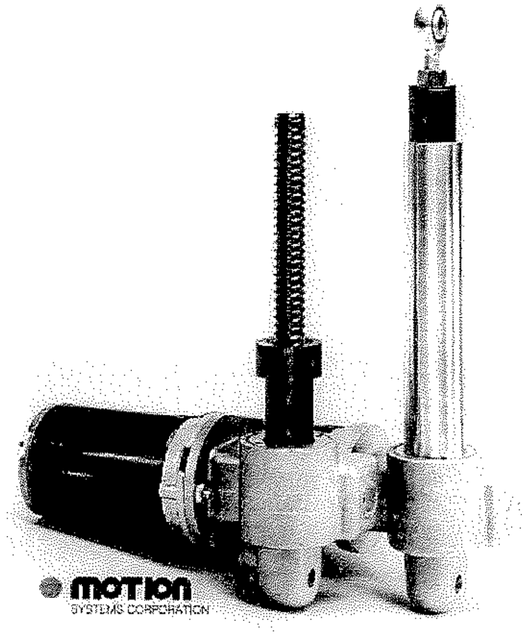


Figure 3.3: Example of a ball-screw actuator.

The final working prototype will, in all probability, use off-the-shelf guide rails. However, since this is a preliminary study considering mainly the functionality and feasibility of the planar platform design, the guide rails are modeled as simple round beam elements. This assumption is made to avoid a mixed real-integer optimization problem, since linear bearings and corresponding guide rails are available only in discrete size increments. Considering the guide rails in this way simplifies the problem greatly in this initial study.

In Section 3.3.2, a technique to tailor the local stiffness matrix of the beam element used to model the sliding guide rails, is explained.

3.2.3 Transverse platform members

The upper transverse platform members (B) are attached to the platform via bearings, allowing rotation relative to the moving platform. These elements represent the upper revolute joint axes. Both the guides and the actuators are fixed to these members. Modeling of the transverse platform members is also accomplished by tailoring the element stiffness matrix (3.4).

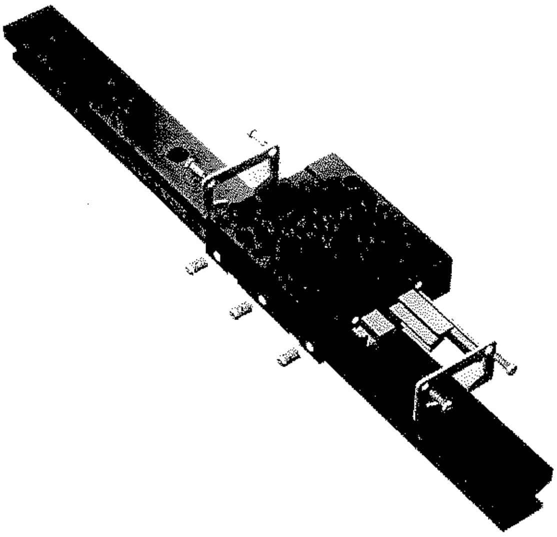


Figure 3.4: Example of a linear bearing guide rail.

3.2.4 Main transverse support struts

The lower transverse supporting struts (C) are supported on either end by bearings which allow rotation about the Z axis. These members represeut the lower revolute joint axes. The actuator driving nuts are supported by these members, and hence the majority of the load is applied at the centers of these elements. This is expected to cause the beams to bend substantially at their center points.

The bearings at either end of the supporting struts can easily be modeled by specifying the correct element boundary conditions.

3.3 Finite elements used in the FEM model

In this section more detail is presented regarding the way in which the FEM model is set up, as well as a brief description of the elements used in the analysis. Additionally, since certain components are rather specialized, it is necessary to explain how the stiffness matrix is tailored to suit the requirements of some of the components in the assembly.

3.3.1 Introduction

The EDSAP finite element analysis program is used to perform the finite element analysis. The original code of Wilson [46] was modified and extended by Groenwold and Stander [12, 13] to incorporate new elements such as the flat shell element used in this study.

For this study it is assumed that the dynamic effects of the moving platform are negligible compared to the static load and to the forces that are applied during machining operations.

Therefore a static finite element analysis is performed at critical platform postures.

3.3.2 Three-dimensional frame element

For more detailed information concerning the formulation of this element, the reader is referred to Appendix B. Figure 3.5 represents a two-noded beam element in its local coordinate system. Each node has six degrees of freedom, three displacement and three rotational, as indicated in the figure.

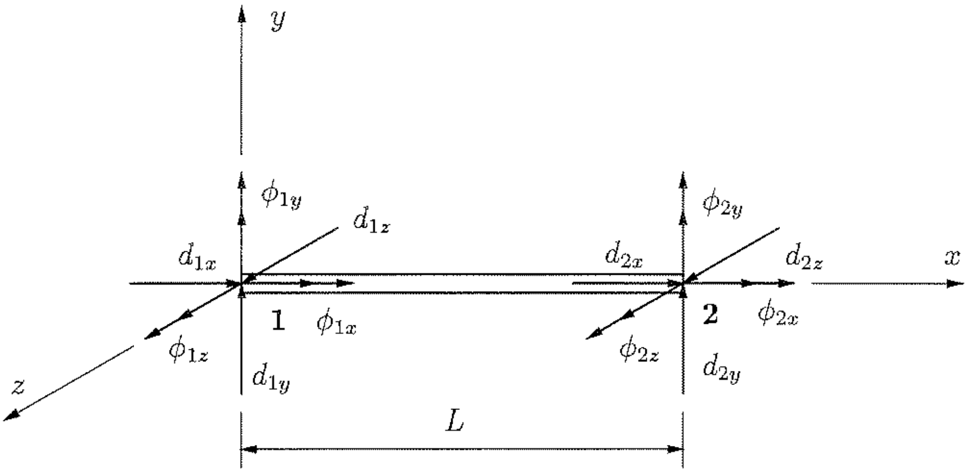


Figure 3.5: Six degree of freedom beam element.

In the local coordinate system the relationship between displacement and applied load is given by:

$$\boldsymbol{f} = \boldsymbol{K}\boldsymbol{q} \tag{3.1}$$

where the components of the displacement vector are given by

$$\boldsymbol{q} = [\ d_{1x} \quad d_{1y} \quad d_{1z} \quad \phi_{1x} \quad \phi_{1y} \quad \phi_{1z} \quad d_{2x} \quad d_{2y} \quad d_{2z} \quad \phi_{2x} \quad \phi_{2y} \quad \phi_{2z}\ ]^T \tag{3.2}$$

and that of the generalized load vector by

$$\boldsymbol{f} = [\ f_{1x} \quad f_{1y} \quad f_{1z} \quad m_{1x} \quad m_{1y} \quad m_{1z} \quad f_{2x} \quad f_{2y} \quad f_{2z} \quad m_{2x} \quad m_{2y} \quad m_{2z}\ ]^T \tag{3.3}$$

The associated stiffness matrix is [47]:

$$\boldsymbol{K} = \left[\begin{array}{cccccccccccc}
AS & 0 & 0 & 0 & 0 & 0 & -AS & 0 & 0 & 0 & 0 & 0 \\
 & a_{z'} & 0 & 0 & 0 & b_{z'} & 0 & -a_{z'} & 0 & 0 & 0 & b_{z'} \\
 & & a_{y'} & 0 & -b_{y'} & 0 & 0 & 0 & -a_{y'} & 0 & -b_{y} & 0 \\
 & & & TS & 0 & 0 & 0 & 0 & 0 & -TS & 0 & 0 \\
 & & & & c_{y'} & 0 & 0 & 0 & b_{y'} & 0 & d_{y'} & 0 \\
 & & & & & c_{z'} & 0 & -b_{z'} & 0 & 0 & 0 & d_{z'} \\
 & & & & & & AS & 0 & 0 & 0 & 0 & 0 \\
 & & & & & & & a_{z'} & 0 & 0 & 0 & -b_{z'} \\
 & & & & & & & & c_{y'} & 0 & b_{y'} & 0 \\
 & & & & & & & & & TS & 0 & 0 \\
 & & & & & & & & & & c_{y'} & 0 \\
Symm & & & & & & & & & & & c_{z'}
\end{array}\right] \tag{3.4}$$

where $AS = AE/L$, L = element length, $TS = GJ/L$, $a_{z'} = 12EI_{z'}/L^3$, $b_{z'} = 6EI_{z'}/L^2$, $c_{z'} = 4EI_{z'}/L$, $d_{z'} = 2EI_{z'}/L$, $a_{y'} = 12EI_{y'}/L^3$, etc.

In order to reduce the stiffness in the local x direction, the terms in the stiffness matrix related to this degree of freedom are multiplied by a scaling factor γ, where $0 \leq \gamma \leq 1$. For this study, γ is taken as zero. In the case of the guide rails, zero axial stiffness is required and is modeled by multiplying all terms in the row corresponding to the axial displacement by γ. In the case of the transverse platform members, zero torsional stiffness is required to model the bearings supporting these elements. Here the elements in the row corresponding to the torsional degree of freedom are multiplied by γ.

3.3.3 Flat shell element with drilling degrees of freedom

A shell finite element [48] refers to an element which is an assemblage or a superimposing of a membrane (plane stress) element and a plate element. The combination of membrane elements with plate elements to form shell elements usually results in a 5-dof per node element. Figure 3.6 schematically shows how such an element is constructed. It may be noted that for the 3-dof plate element and 2-dof membrane element there is no stiffness component in the θ_z direction. More detail regarding the assemblage of shell elements in this way may be found, for example, in the work of Zienkiewicz and Taylor [49].

The shell element used in this study to model the platform slab is an assembly of the modified QCnD membrane element [12, 13], based on the element of Ibrahimbegovic *et al.* [11], and the Bathe-Dvorkin [50] assumed strain plate element. The resulting shell element is denoted QCnD-SA, with n indicating the number of integration points used in the membrane component. Figure 3.7 represents this shell element with the additional in-plane drilling degrees of freedom.

Generally a membrane element with drilling degrees of freedom is not required in a shell assembly. Alternative approaches can work in locally defined systems with 5-dof per node. Small artificial stiffnesses associated with the in-plane degrees of freedom can then be incorporated. These artificial stiffnesses are modeled with an equivalent "imaginary" torsional

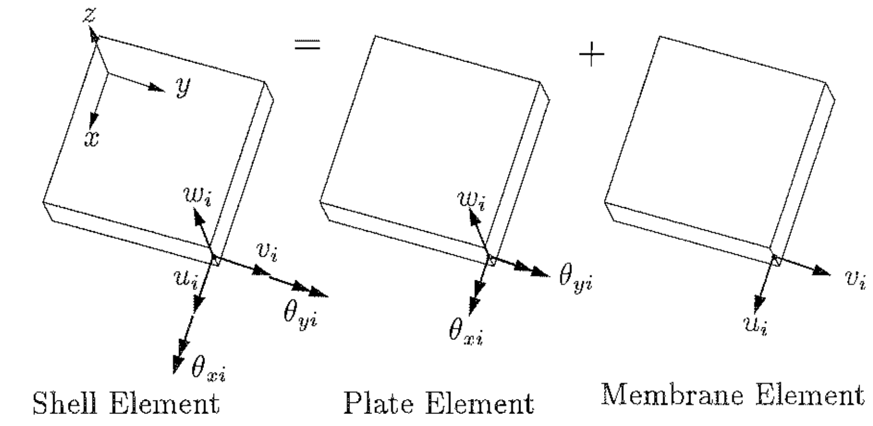


Figure 3.6: Regular shell element.

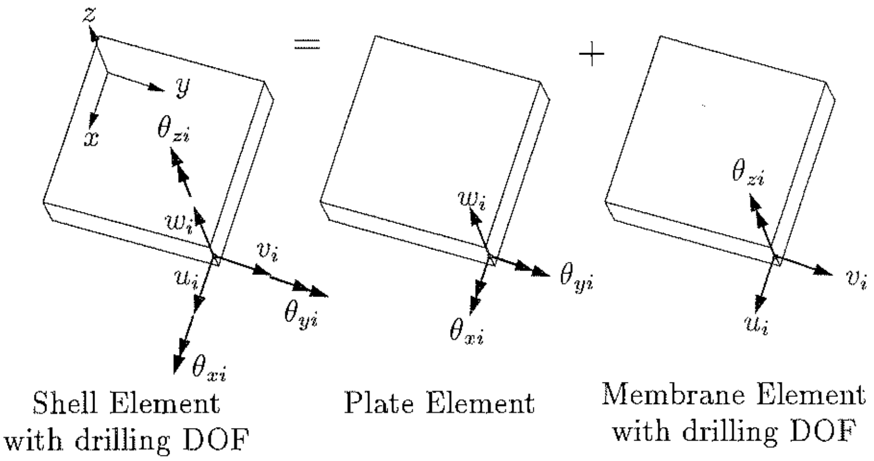


Figure 3.7: Shell element with drilling degrees of freedom.

spring at each node. The presence of drilling degrees of freedom however, allows for beam-shell connections and enriches the calculated displacement field. Hence they enhance the element accuracy.

The force-displacement relationship for the QCnD-SA shell element may be represented by

$$\begin{bmatrix} (\boldsymbol{K}_m+\boldsymbol{P}) & \boldsymbol{0} & \boldsymbol{K}_{mh} \\ & (\boldsymbol{K}_b+\bar{\boldsymbol{K}}_s) & \boldsymbol{0} \\ symm & & \boldsymbol{K}_h \end{bmatrix} \boldsymbol{q} = \boldsymbol{f} \tag{3.5}$$

for each of the corner nodes, $i = 1, 2, 3, 4$ the shell nodal displacements and loads are given by

$$\boldsymbol{q}_i = [u_i \;\; v_i \;\; w_i \;\; \theta_{xi} \;\; \theta_{yi} \;\; \theta_{zi}]^T \tag{3.6}$$

$$\boldsymbol{f}_i = [U_i \;\; V_i \;\; W_i \;\; M_{xi} \;\; M_{yi} \;\; M_{zi}]^T \tag{3.7}$$

respectively. Further information regarding these specific elements may be found in Appendix C.

3.4 Model boundary conditions

3.4.1 Model restraints

In Section 3.2 the components of the platform, as well as the way in which each component is modeled, were discussed. The only restraint that was specifically specified in the EDSAP input file pertains to the bearings on either end of the main transverse support struts (denoted I in Figure 3.2).

These bearings are easily modeled, by simply specifying that the nodes coinciding with the bearing are free to rotate about the Z axis and constrained in all other degrees of freedom.

It is therefore assumed that the frame to which the platform is attached is rigid and that the amount that the frame deflects is negligible compared to the deflection of the platform members themselves. This is generally a good assumption since a rigid frame is easily manufactured and fixed to a solid foundation.

3.4.2 Loading

There are two sets of forces that act on the platform, namely the tool forces and the body forces due to the weight of the elements. Note that gravity in the model acts in the X direction, since the stationary platform will be mounted vertically on an existing (horizontal) milling machine as shown in Figure 3.1. Of course, if a horizontally mounted platform is considered it is a simple matter to change the direction in which the gravitational acceleration acts.

Since the model is discretized into a number of small finite elements, the weight contribution of each element needs to be calculated individually. These body forces then need to be applied at the element nodes.

Since beam elements with Hermitian shape functions are used, evenly distributed loads equate to equivalent nodal loads that include moments. In the case of the shell elements the nodal moments are not present since separate interpolation functions are used for displacements and rotations.

Previously Smit [51] studied the forces that act on a workpiece while actual milling operations are taking place. These forces are a function of a number of factors including the cut speed, the rotational speed of the tool, the depth of the cut, the type of material being machined, the type of lubrication used and many other factors. He shows that for a typical application the forces acting on the structure in the X and Z directions are 1600N and 1800N respectively, and in the negative Y direction 440N. These forces act at the working point (P) on the moving platform. More detailed information regarding these calculations may be found in Appendix G.

The weight of the workpiece is not explicitly taken into account. It is expected to be relatively small and its magnitude will vary from application to application.

3.5 Preliminary finite element results

In this section the results of the finite element analysis carried out on the initially proposed design are discussed. These results are used to critically analyze the design and to suggest design improvements. The way in which the structure deforms under load gives an important insight into potentially weak links in the design, and points to possible design improvements.

A representative design is chosen that can possibly be used for machining operations. This design is roughly based on a scaled version of the platform that Haug *et al.* [52] studied, and for which the workspace has been calculated and documented for specific leg length constraints. This workspace can be used to find points where stiffness is expected to be low, or deflection high, for the given load. The chosen design is shown in Figure 3.8 in both the loaded (thin lines) and unloaded (bold lines) conditions, and in four different views (a-d). (All dimensions in Figure 3.8 are in meters.)

3.5.1 Deformation analysis

In order to study the platform deformation under loading, elements in the model need to be provided with material properties and section profiles. For this analysis, all elements are assumed to be constructed from mild steel. All beam elements are assumed to have a diameter of 40mm and the moving platform slab is chosen to be 40mm thick. This model is sufficiently realistic to provide information about the way in which the platform is expected to deflect under load.

A finite element analysis is carried out on this model and the displacements of each node in the model are calculated and used to plot Figure 3.8. In the figure, displacements are amplified 150 times and are superimposed over the original model. For clarity only the beam elements are plotted.

The displacements of the nodes point to certain design shortcomings and provided valuable insight for the later analysis of the optimization results.

The top view (d) and side view (b) of the platform displacement reveal that the magnitude of the in-plane X displacement, compared to that of the out of plane Z displacement, is surprisingly high considering that parallel platforms are known to be very stiff in their work plane. However, this result is partially explained by the fact that gravity acts in the X direction.

The front (c) and top (d) views of the platform show the extent to which the lower transverse supporting struts deflect in the Y and X directions respectively. These large deflections are due to the fact that the struts are supported only at the end points by bearings. This means that the bending moments at the center of the supporting struts are large, and that the members can relatively easily bend at their centers. The transverse support struts should thus be supported by bearing nearer to their centers in order to reduce this deformation.

The top view (d) shows how the upper transverse members bend substantially at their center points. These deformations can be eliminated by fixing the upper transverse members to the platform at their centers. However, careful consideration will need to be paid to the

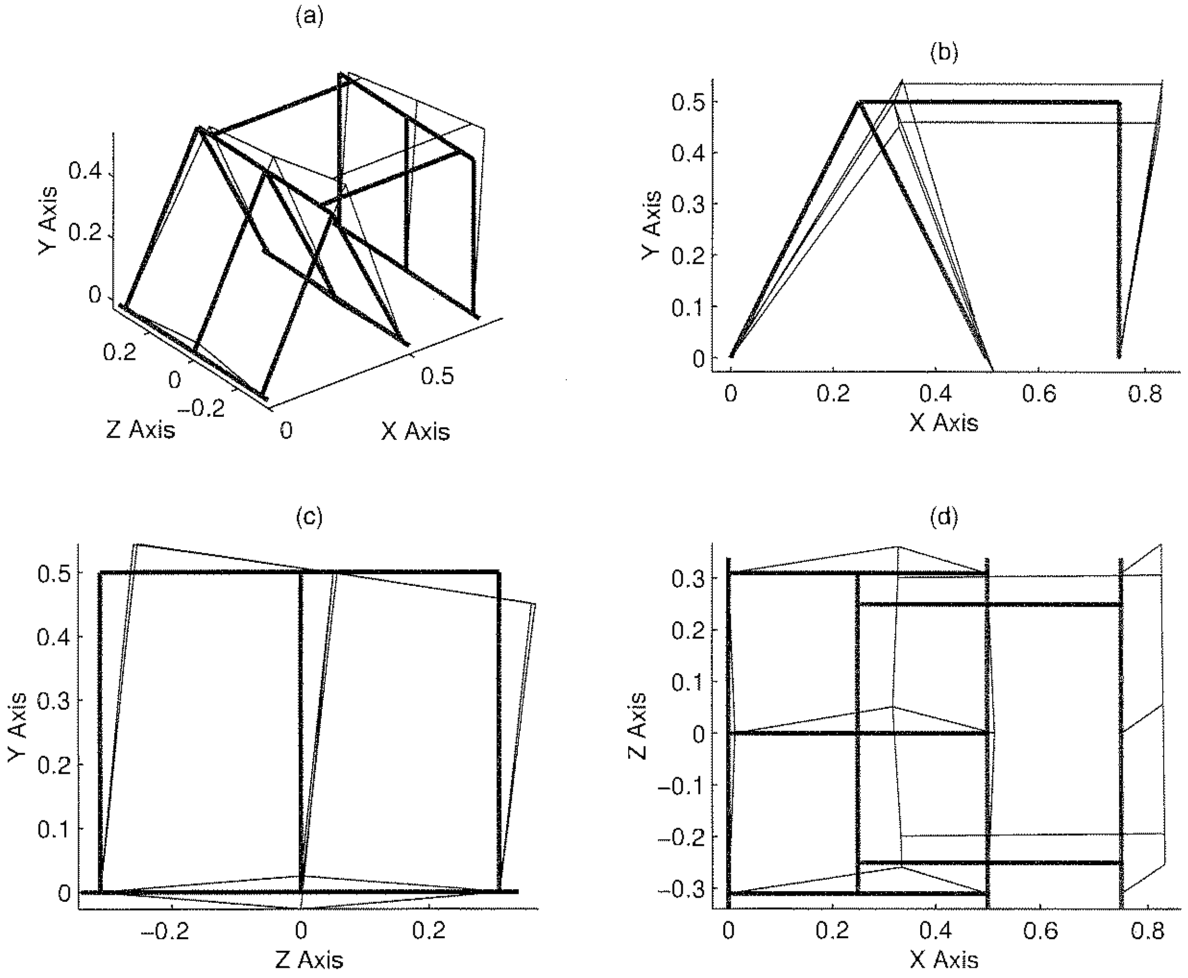


Figure 3.8: Four views (a-d) of deformation of planar platform due to machine tool loading and own weight.

practical design of this connection point.

The most remarkable characteristic of the displacement behavior is the way in which the platform rotates about the X axis as depicted in view (c). This rotation is largely due to the fact that the guide rails are not functioning as was initially intended. Since the guide rails are allowed to move freely in their axial direction, rotation is not countered effectively. The only stiffness preventing the Z displacements of the platform, is that associated with the bending stiffness of the actuators and the guide rails.

3.6 Design sensitivity studies

For improvement in the design via formal mathematical optimization, it is necessary to find a workspace point that represents a worst case with respect to displacement. Two factors that are important in this respect are the Y coordinate of the working point (Y_p) and the width of the platform. Again for this study all beam elements are assumed to have a diameter of 40mm and the platform's thickness is selected as 40mm.

In order to determine a representative point under the given conditions, the finite element analysis is carried out for the Y_p coordinate of the working point stepped up from its lowest to its highest value in the workspace, keeping the orientation at zero degrees throughout and for a fixed width of 0.5 m. The X_p coordinate of the working point is fixed at 0.5 m, as prescribed by the workspace calculated by Haug *et al.* [52]. This does not strictly correspond to the situation in the workspace calculation of Haug *et al.* since in their case the orientation does change along the boundary of the workspace for the specific maximum and minimum actuator lengths that are used.

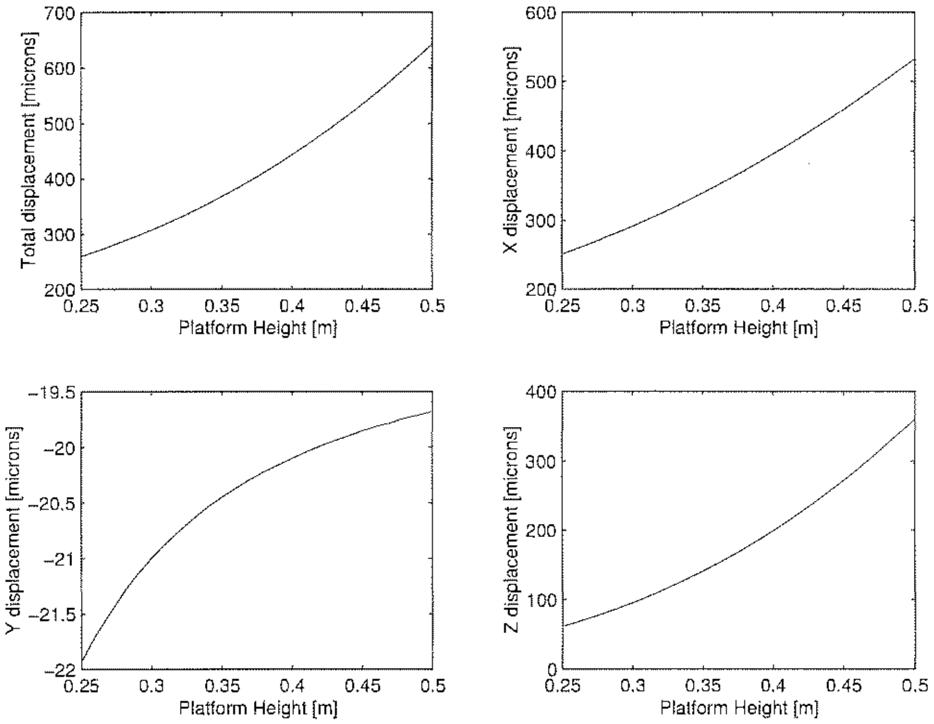


Figure 3.9: Sensitivity of displacement to the height of the working point at a fixed width of 0.5 m.

From Figure 3.9 which depicts various displacements as a function of height (where height refers to the value of Y_p), it is noticeable that the total displacement increases as Y_p increases. This results is rather predictable as the structure is made up of mainly beam elements in bending.

Only the Y coordinate of the displacement does not increase as the height is increased. However, the magnitude of this displacement is an order of magnitude smaller than that of the components which increase. Therefore the worst case with respect to the working point deflection occurs at the maximum height of the working point as expected.

A similar study is carried out for the working point at a fixed height of 0.5 m in order to determine how the platform reacts to a change in width. Again the platform is stepped from

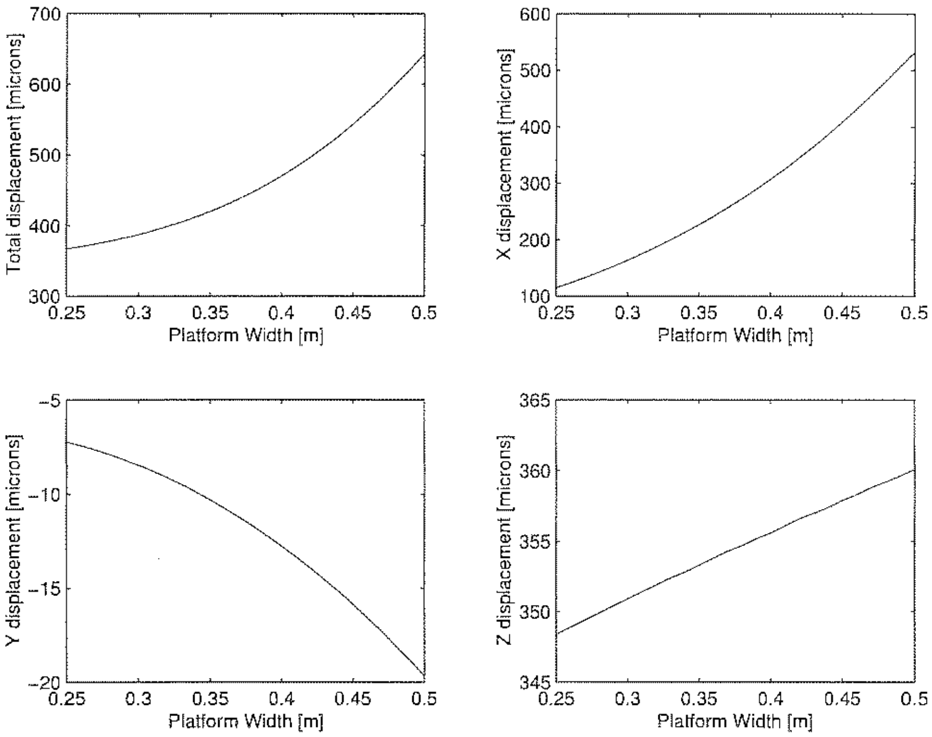


Figure 3.10: Sensitivity of displacement to the platform width at a fixed height of 0.5 m.

a relatively narrow width to a reasonably large width.

It can be seen from Figure 3.10, which plots corresponding displacements as functions of width, that the total displacement increases as the platform width is increased. This result seems counter intuitive, since one expects the deflection to decrease as width increases since material is being added further from the plane of symmetry, which corresponds to a "neutral axis" of bending.

However, this argument does not hold since there is a coupling between the various displacement components due to the asymmetric loading. Furthermore, the X displacement increases with an increase in width for two reasons. Firstly, the wider the platform becomes the more material, and thus the greater weight that needs to be supported. Secondly the lengths of the lower support struts are also increased with an increase in width. Therefore not only is there a higher load, but the bending moments on the lower struts are increased.

For practical reasons of utility the platform width will be fixed at 0.5 m and the analysis will be performed for a posture with the platform orientation at zero degrees. The working point P will be fixed at $Y_p = 0.5$ m and $X_p = 0.5$ m.

Chapter 4

Optimization of an original concept design

Mathematical optimization techniques are used to evaluate the existing concept design of the planar parallel platform. Firstly, it must be determined whether a feasible design exists, and if a feasible design indeed exists, the design is to be optimized with respect to weight.

For the computed optimization results, the genetic algorithm due to Carrol [14] is used as both a micro GA and a full GA. Two different gradient-based algorithms, both originated by Snyman, are also used. They are the leap-frog algorithm for constrained optimization [15] and the Dynamic-Q algorithm [16]. The performance of the GA's is compared to that of the gradient-based methods.

4.1 Formal optimization formulation

In Section 3.6 a point in the workspace with a low stiffness was determined. The optimization is performed with the platform in this extreme posture. More precise detail of the configurational settings are tabulated in Table 4.1, with reference to the annotation specified in Figure 4.1. The only permanently fixed configurational setting is the platform width, since the other (in plane) settings may be adjusted in order to meet other specific workspace requirements as described in Section 2.2.8. The software written for this study thus needs to be compatible with this existing work.

The configurational settings may be represented by the parameter vector

$$\boldsymbol{U} = [X_0 \;\; Y_0 \;\; X_1 \;\; X_2 \;\; X_p \;\; Y_p \;\; l_p \;\; \phi \;\; w]^T \tag{4.1}$$

which remains fixed during a given optimization process.

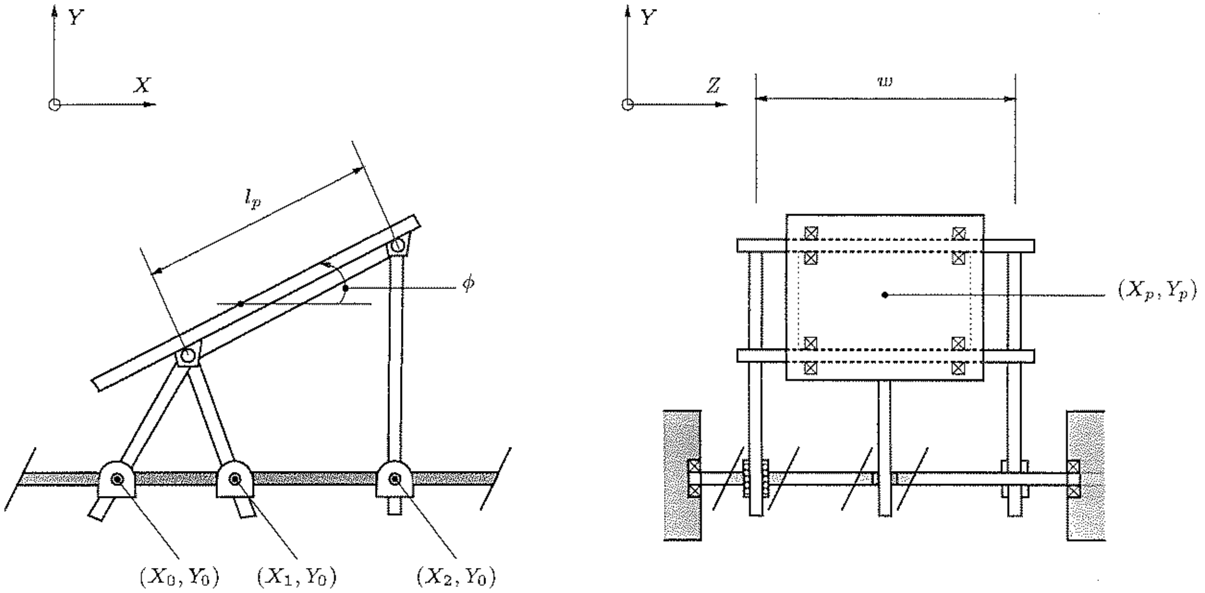


Figure 4.1: Configuration of planar parallel platform.

Configurational setting	Value [m]
X_0	0.00
Y_0	0.00
X_1	0.50
X_2	0.75
X_p	0.50
Y_p	0.50
l_p	0.50
ϕ	0.00
w	0.50

Table 4.1: Configurational setting for optimization.

4.1.1 Design variables

The optimization is carried out with respect to the sizes of the various components, i.e. their respective dimensions are taken as the design variables. In particular, the six components of the vector of design variables $\boldsymbol{x} = [x_1, x_2, ..., x_6]^T$, considered here are the diameters of the five different components modeled as beam elements, and the thickness of the platform.

No a prioir attention is paid to the sensitivity of the cost function to the design variables before proceeding with the optimization, in order to reduce the number of design variables used. Indeed, the maximum number of design variables are used in order test the performance of the optimization algorithms.

A precise description of the design variables that are used in the optimization are listed below

- x_1 – actuator diameter,
- x_2 – guide diameter,
- x_3 – bottom transverse support strut diameter,
- x_4 – upper transverse platform member diameter,
- x_5 – lateral platform member diameter, and
- x_6 – platform thickness.

4.1.2 Constraints

For the initial evaluation of the proposed design, only bounds on the variables and a simple constraint on the maximum allowable displacement of the working point are specified. Two cases are considered regarding the variable bounds. The first is a lenient case in which the variables are effectively unbounded. The second is a more strict set of constraints, used to determine what accuracy the platform design can realistically achieve.

The lenient set of constraints are required since most simple binary GA's have a discrete number of possible variable values, i.e. a maximum and minimum bound and a number of (equally spaced) possibilities. Inherently therefore unbounded variables are impossible to define. In order to compare the results of the GA's with the two gradient-based methods, for this large variable space, similar constraints are used for both methods.

The lenient side constraints are of the form

$$\hat{k}_i \leq x_i \leq \check{k}_i \ , \qquad i = 1, 2, \ldots, 6 \tag{4.2}$$

where $\hat{k}_i$ refers to the lower bound of x_i, and $\check{k}_i$ refers to the upper bound. The numerical values of the lower and upper bound respectively are given in millimeters below:

$$\begin{aligned} \hat{\boldsymbol{k}} &= [1 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 1]^T \\ \check{\boldsymbol{k}} &= [500 \;\; 500 \;\; 500 \;\; 500 \;\; 500 \;\; 500]^T \end{aligned} \tag{4.3}$$

The lower and upper constraint functions (required for the standard ≤ 0 formulation), are respectively given by

$$\begin{aligned} \hat{g}_i &= \hat{k}_i - x_i \ , \qquad i = 1, 2, \ldots, 6 \\ \check{g}_i &= x_i - \check{k}_i \ , \qquad i = 1, 2, \ldots, 6 \end{aligned} \tag{4.4}$$

The set of representative and relatively strict constraints are used to determine the minimum deflection that is realistically attainable from the concept design. The set of strict side constraints are written as

$$\hat{K}_i \le x_i \le \check{K}_i \ , \qquad i = 1, 2, \ldots, 6 \tag{4.5}$$

Here, the numerical values (again in millimeters) of the lower and upper bounds are respectively specified by the vectors

$$\begin{aligned} \hat{\boldsymbol{K}} &= [10 \quad 10 \quad 30 \quad 30 \quad 30 \quad 30]^T \\ \check{\boldsymbol{K}} &= [60 \quad 50 \quad 100 \quad 50 \quad 50 \quad 40]^T \end{aligned} \tag{4.6}$$

The corresponding set of lower and upper constraint functions, required for the standard ≤ 0 formulation, may now be specified as

$$\begin{aligned} \hat{G}_i &= \hat{K}_i - x_i \ , & i &= 1, 2, \ldots, 6 \\ \check{G}_i &= x_i - \check{K}_i \ , & i &= 1, 2, \ldots, 6 \end{aligned} \tag{4.7}$$

Additionally, a displacement constraint is used to ensure sufficient stiffness. The stiffness of the platform has a direct influence on the tool life for machining operations. If the platform is sufficiently stiff, undesirable phenomena such as tool chatter are less likely to occur. Furthermore, the global stiffness is directly related to the accuracy with which machining operations are performed. For these reason it is necessary to constrain the displacement of the working point. The constraint on the displacement of the working point may be written as

$$\bar{g}(\boldsymbol{x}) = d(\boldsymbol{x}) - \bar{d} \le 0 \tag{4.8}$$

where $d(\boldsymbol{x})$ is the implicit displacement function of the working point (P) computed via FEM for any given design vector $\boldsymbol{x}$, and where $\bar{d}$ is the maximum allowable displacement of the working point. Depending on the machining requirements various target tolerances for $\bar{d}$ may be prescribed. For the rough milling of molds a sufficient design should be obtained if $\bar{g}(\boldsymbol{x})$ is satisfied with $\bar{d} = 1$ mm. However, since the current aim is to perform practical machining operations as accurately as possible, the limit on $\bar{d}$ is set to as little as 10 μm for the force vector prescribed by Smit [51] (see Appendix G). The computed displacements do not take into account factors such as play in joints and bearing stiffnesses [4].

In order to investigate the capabilities of the design, three different values of $\bar{d}$ are selected, namely 1000 μm, 100 μm and 10 μm respectively. The optimization process is then successively carried out for the different displacement constraints, with this constraint becoming progressively stricter. These results are also used to compare the performances of the different optimization methods.

The lenient constraint function vector (with components specified by (4.4) and (4.8)), may be written as

$$\boldsymbol{g}(\boldsymbol{x}) = [\hat{\boldsymbol{g}}(\boldsymbol{x}) \quad \check{\boldsymbol{g}}(\boldsymbol{x}) \quad \bar{g}(\boldsymbol{x})]^T \tag{4.9}$$

while the strict constraint function vector (components (4.7) and (4.8)) is denoted by

$$\boldsymbol{G}(\boldsymbol{x}) = [\hat{\boldsymbol{G}}(\boldsymbol{x}) \quad \check{\boldsymbol{G}}(\boldsymbol{x}) \quad \bar{g}(\boldsymbol{x})]^T \tag{4.10}$$

4.1.3 Cost function

Reducing the weight of a parallel manipulator used in machining applications has many advantages. A low weight structure is more maneuverable, resulting in faster material removal rates. It is also less expensive and the need for heavy duty foundations is eliminated. For these reasons the weight of the platform structure is minimized with respect to the selected design variables and subject to the prescribed constraints.

The weight is calculated by summing all the weights (in kg) of the individual finite elements, and may be written as

$$F(\boldsymbol{x}) \quad = \quad \sum_{i=1}^{m}(w_i) \tag{4.11}$$

where w_i represents the weight of finite element i, and m represents the total number of elements in the model.

The optimization problem may now be expressed in the general and standard form:

$$\begin{aligned} \min_{\boldsymbol{x}} F(\boldsymbol{x}) \;, \qquad & \boldsymbol{x} = [x_1 \;\; x_2 \;\; x_3 \;\; x_4 \;\; x_5 \;\; x_6]^T \\ \text{such that} \quad & \boldsymbol{g}(\boldsymbol{x}) \leq \boldsymbol{0} \text{ or } \boldsymbol{G}(\boldsymbol{x}) \leq \boldsymbol{0} \end{aligned} \tag{4.12}$$

Here $\boldsymbol{g}(\boldsymbol{x})$ and $\boldsymbol{G}(\boldsymbol{x})$ are the two distinctly different constraint function sets given by (4.9) and (4.10) respectively.

A penalty function approach is used to solve the above constrained optimization problem. The penalty function to be minimized for the lenient constraint set is formally defined as

$$\begin{aligned} P(\boldsymbol{x}) \;, \qquad & \boldsymbol{x} = (x_1 \;\; x_2 \;\; x_3 \;\; x_4 \;\; x_5 \;\; x_6)^T \\ \text{where } P(\boldsymbol{x}) = & F(\boldsymbol{x}) + \sum_{i=1}^{13} \beta_i g_i^2(\boldsymbol{x}) \\ \text{and } \beta_i = & \begin{cases} 0 & \text{if } g_i(\mathbf{x}) \leq 0 \\ \mu_i & \text{if } g_i(\mathbf{x}) > 0 \end{cases} \end{aligned} \tag{4.13}$$

and the penalty parameter $\mu_i \gg 1$ is prescribed.

4.2 Implementation of optimization methodology

Figure 4.2 illustrates the effect on both the cost function F and the penalty function P with respect to a change in variable x_j near the lower bound. When the lower bound is violated, the function value is penalized to ensure that this area is avoided. The figure also shows that x_j cannot be negative since this value is physically undefined.

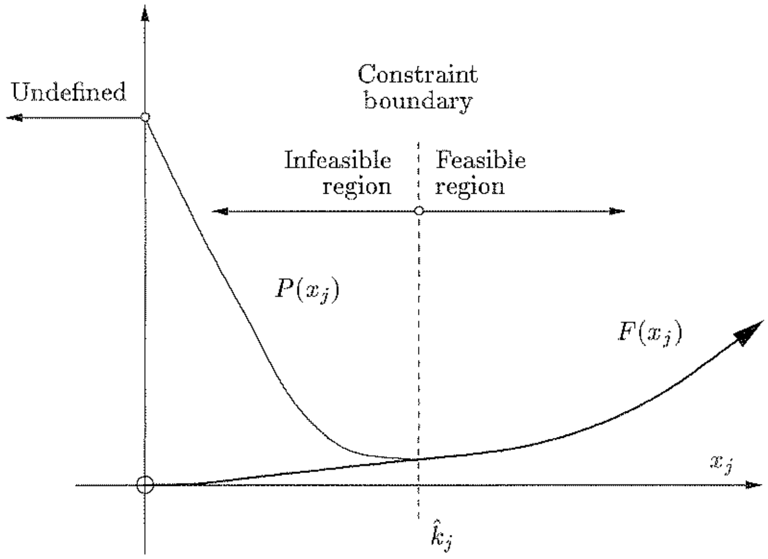


Figure 4.2: Graphical illustration of violation of a lower bound constraint.

In practice, even though a penalty function approach is used by the gradient-based optimization algorithms employed here, and the penalty function $P(\boldsymbol{x})$, is drastically increased when a constraint is violated, it does not prevent the algorithms from sporadically entering infeasible regions. If a variable value becomes too low, the FEM analysis becomes inaccurate due to the poor scaling of the stiffness matrix. Therefore, these badly scaled regions, as well as the undefined regions, need to be avoided.

To this end, a point denoted here by $\bar{x}_j$ (see graphical illustration in Figure 4.3), is determined such that, even though the side constraint is violated, the stiffness matrix is not badly scaled. Now, if x_j, predicted by the algorithm, lies in the undesirable region it is artificially moved to the point $\bar{x}_j$ where the function value is penalized, but can still be accurately evaluated. Since the constraint is violated at this point, the large slope will force the algorithm back to the feasible region.

Figure 4.4 gives a diagrammatic representation of the basic optimization procedure. The constraint and cost functions for the initial choice of design variables $\boldsymbol{x}^0$, as well as the gradient vector, $\boldsymbol{\nabla}\mathrm{P}(\boldsymbol{x})$, of the associated penalty function, are evaluated. The design variables are then systematically changed by the selected algorithm, and the effect of these changes on the constraint and cost functions is monitored after each iteration. The process of changing design variables and monitoring the constraint and cost functions is continued until a stopping criterion is satisfied. This design is then taken as the optimum design, $\boldsymbol{x}^*$.

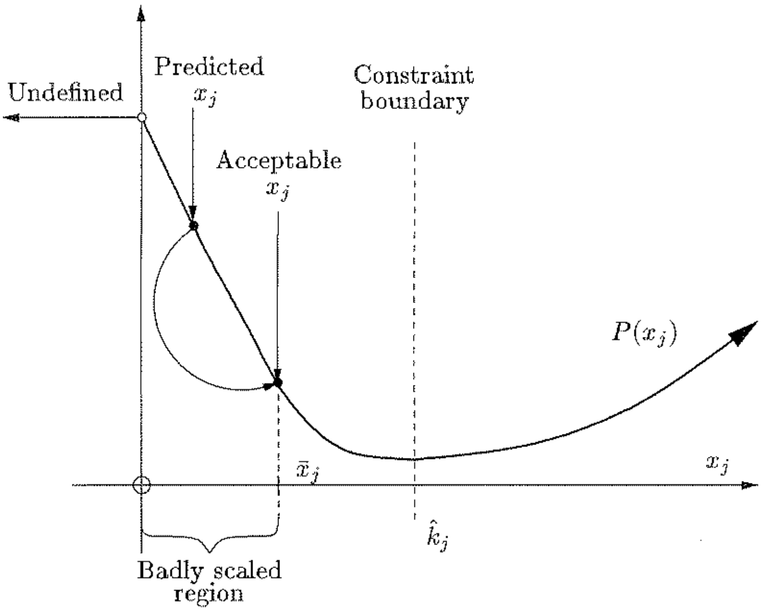


Figure 4.3: Graphical illustration of scheme aimed at avoiding badly scaled regions.

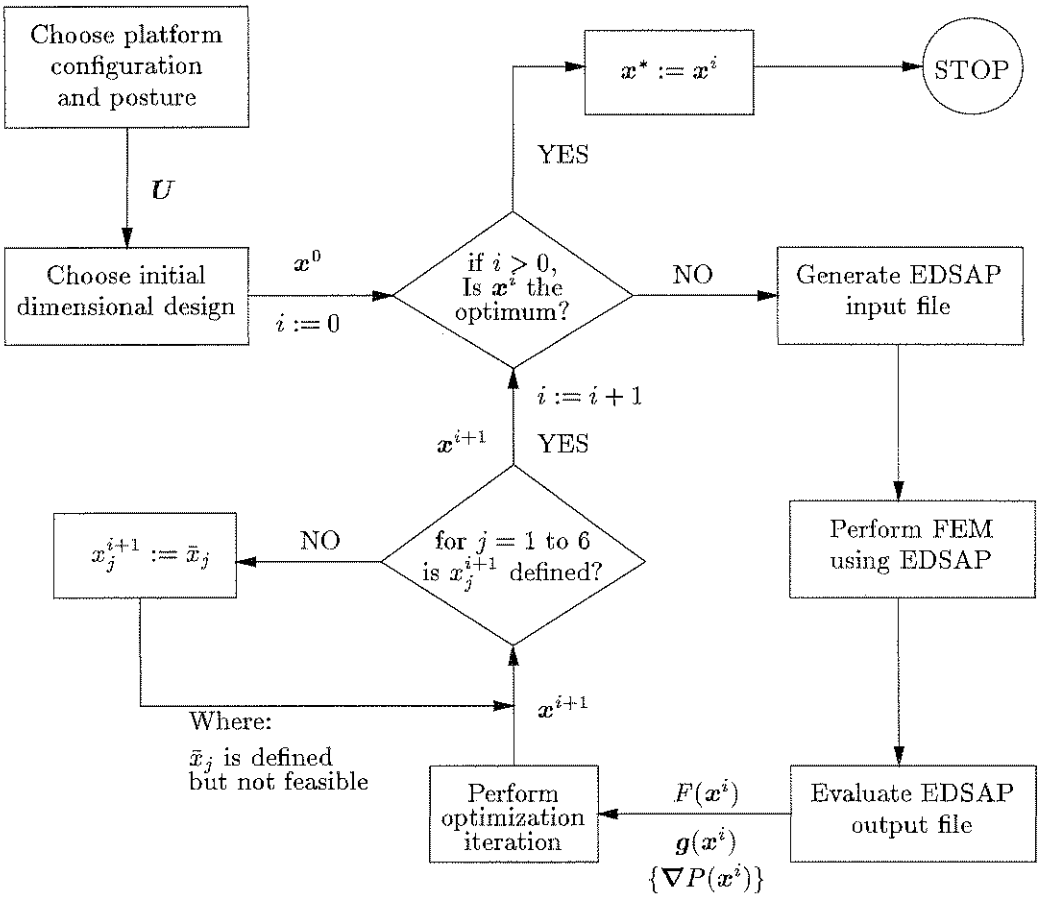

Choose platform configuration and posture
U
Choose initial dimensional design
x^0
$i := 0$
if $i > 0$, Is x^i the optimum?
YES
$x^* := x^i$
STOP
NO
Generate EDSAP input file
Perform FEM using EDSAP
Evaluate EDSAP output file
$F(x^i)$
$g(x^i)$
$\{\nabla P(x^i)\}$
Perform optimization iteration
x^{i+1}
for $j = 1$ to 6 is x_j^{i+1} defined?
YES
$i := i + 1$
x^{i+1}
NO
$x_j^{i+1} := \bar{x}_j$
Where:
$\bar{x}_j$ is defined but not feasible

Figure 4.4: Diagrammatic representation of the optimization procedure.

4.2.1 The genetic algorithm of Carrol

Genetic algorithms search the solution space of a function through the use of simulated evolution, i.e. the survival of the fittest strategy. In general the fittest individuals of any population tend to reproduce and survive to the next generation, thus improving successive generations. However, inferior individuals can, by chance, survive and also reproduce. Genetic algorithms have been shown to solve linear and nonlinear problems by exploring regions of the state space and exploiting promising areas through mutation, crossover and selection operations applied to individuals in the population.

In a micro GA relatively small populations are selected. These small populations quickly converge to a solution, which is stored and usually survives, while re-birth occurs in the rest of the population. This process is repeated a number of times until a suitable global optimum is found. Hence, rebirth replaces mutation as the operator that ensures genetic diversity. It would be futile to suggest a "universally best" GA. Indeed this quest contradicts the "free-lunch" theorem [53, 54]. Nevertheless, a comparison between mutation and rebirth for a specific class of problems is in order and of interest.

The model of the platform is not very complex by modern standards. The analysis (or function evaluation) is evaluated relatively quickly and takes less than a second to be computed. For this reason no complicated stopping criteria are used. The algorithm is simply allowed to run for 5000 iterations, unless a specified number of generations pass with no improvement in best function value.

For more detail about genetic algorithms, the reader is referred to Appendix F.

4.2.2 The dynamic trajectory "leap-frog" optimization method

LFOPC of Snyman [15, 55, 56] is a well-established constrained optimization code. This code uses a very robust gradient descent optimization algorithm which handles discontinuities that may occur in the gradients, as well as noise in the constraint and objective function with ease. Considering the fact that the function values are computed via a numerical method and that gradients are computed by forward finite differences, the occurrence of such difficulties is a distinct possibility here.

The unconstrained form of the Leap-Frog OPtimizer (LFOP) determines the minimum of a function $F(\boldsymbol{x})$ by considering the associated dynamic problem of the motion of a particle of unit mass in an n-dimensional conservative force field, where the potential energy of the particle at a point $\boldsymbol{x}(t)$ at time t is represented by $F(\boldsymbol{x})$. The algorithm thus requires the solution the equations of motion:

$$\ddot{\boldsymbol{x}}(t) = -\nabla F(\boldsymbol{x}(t)) \tag{4.14}$$

subject to the initial conditions

$$\boldsymbol{x}(0) = \boldsymbol{x}^0; \quad \dot{\boldsymbol{x}}(0) = \boldsymbol{v}^0 \tag{4.15}$$

During the motion an interfering strategy is implemented by means of which energy is systematically extracted from the particle. This ensures that the particle trajectory converges to a local minimum of $F(\boldsymbol{x})$.

For constrained problems, the unconstrained trajectory method is applied to a penalty function formulation of the original constrained problem [15].

More detail about the LFOP and LFOPC algorithms can be found in Appendix D.

4.2.3 The Dynamic-Q algorithm of Snyman

Dynamic-Q [16, 57] constructs and solves successive sub-problems in which the constraint and cost functions are approximations to the actual constraint and cost functions. For each approximate sub-problem *spherical quadratic* functions are constructed using function and gradient information of the actual constraint and cost functions, evaluated at the solution points of the previous two sub-problems. In this manner solutions to a sequence of sub-problems are obtained which, in practice and under relatively general conditions, invariably converges rapidly to the solution of the original problem. The simple and computationally inexpensive sub-problems are solved by the LFOPC algorithm.

This algorithm has been successfully applied to various engineering problems [58, 59, 60] where the cost or constraint functions are computationally expensive. More information regarding this algorithm is presented in Appendix E.

4.3 Optimization results

In this section the results of the optimization that is carried out on the platform are discussed. The results for the gradient-based methods are compared with those of the GA's.

It should be noted that finite forward differences are used to calculate the gradient vectors where required. For the gradient-based methods, therefore, after N algorithm iterations, $N(n+1)$ finite element analyses are carried out, where n represents the number of variables. Thus, in this case, one iteration requires seven function evaluations.

Displacements, calculated by the finite element method, are only made available correct to four significant figures in the EDSAP output file. Therefore, the magnitude of the displacement at the working point cannot be determined precisely. This accounts for the presence of numerical noise in the computed displacement function. A relatively large finite difference increment (0.1 mm) is therefore used to compensate for this noise.

In Appendix H the computed sensitivities of the objective and constraint functions to the various variables are presented and discussed. In order to determine these sensitivities, 20 finite difference increments are used on either side of the desired point, and the effect on the relevant function determined. This process is repeated for each design variable, while the other variables are kept constant.

The initial sets of design variables (starting point) for the two gradient-based algorithms are given below. In the case of the lenient constraint functions given by (4.9), an initial design variable set is chosen as

$$\boldsymbol{x}_l^0 = [100,\ 100,\ 100,\ 100,\ 100,\ 100]^T \tag{4.16}$$

In the case of the strict set of constraint functions given by (4.10), some of the aforementioned initial variable choices are infeasible. Thus, the variables which would begin from an infeasible point are moved to a feasible point (at their upper variable bound). The resulting initial set of design variables for the strict set of constraint functions is given by

$$\boldsymbol{x}_s^0 = [60,\ 50,\ 100,\ 50,\ 50,\ 40]^T \tag{4.17}$$

In the tables which follow, containing the information regarding the terminal values obtained for each of the optimization methods, the following abbreviations are used to indicate the applicable criterion for stopping.

- Dmax - Dynamic-Q terminated on maximum number of steps. (max = 300 iterations),
- D$f$tol - Dynamic-Q terminated on function value if the change in function value $\leq \varepsilon_f = 10^{-4}$ kg,
- D$x$tol - Dynamic-Q terminated on step size if the change in $\boldsymbol{x} = ||\Delta\boldsymbol{x}|| \leq \varepsilon_x = 10^{-6}$ m,
- Lmax - LFOPC terminated on maximum number of steps per phase. (max = 300 iterations),
- L$x$tol - LFOPC terminated on step size if change in $\boldsymbol{x} = ||\Delta\boldsymbol{x}|| \leq \varepsilon_x = 10^{-6}$ m,
- FGmax - Full GA terminated on maximum number of steps (max = 5000 function evaluations),
- FG20 - Full GA terminated after 20 generation with no change in best $F(\boldsymbol{x})$,
- MGmax - Micro GA terminated on maximum number of steps (max = 5000 function evaluations), and
- MG100 - Micro GA terminated after 100 generation with no change in best $F(\boldsymbol{x})$.

The stopping criteria for the gradient-based methods are relatively strict. In order to terminate on function value, a change of less than 0.1 g is required, and a $||\Delta\boldsymbol{x}||$ of less than 1 μm for termination on step size. Therefore, it is a notable achievement if these algorithms terminate on their own criteria considering the distinct possibility of numerical noise. More lenient stopping criteria could have been used, however these strict criteria better test the algorithms.

4.3.1 Optimization of original design for constraint set $g(x)$ and $\bar{d} = 1000\ \mu$m.

	Dynamic-Q	LFOPC	Full GA	Micro GA
$F(\boldsymbol{x}^*)$ [kg]	66.901	66.901	69.709	87.375
$\bar{g}(\boldsymbol{x}^*)$ [μm]	-1.477e-01	-8.215e-02	-8.097e+00	-1.233e+01
Function evaluations	1113	2247	5000	5000
x_1^* [mm]	48.748	48.674	49.826	34.202
x_2^* [mm]	1.000	1.000	1.977	32.248
x_3^* [mm]	40.760	40.913	38.108	49.826
x_4^* [mm]	28.816	28.770	33.225	32.248
x_5^* [mm]	24.971	24.859	23.460	20.530
x_6^* [mm]	5.519	5.504	6.859	5.883
Termination type	D$f$tol	Lmax	FGmax	MGmax

Table 4.2: Original design with constraint set $\boldsymbol{g}(\boldsymbol{x})$ and $\bar{d} = 1000\ \mu$m: Terminal values.

The results for this optimization run are tabulated in Table 4.2 and the convergence histories are depicted in Figure 4.5.

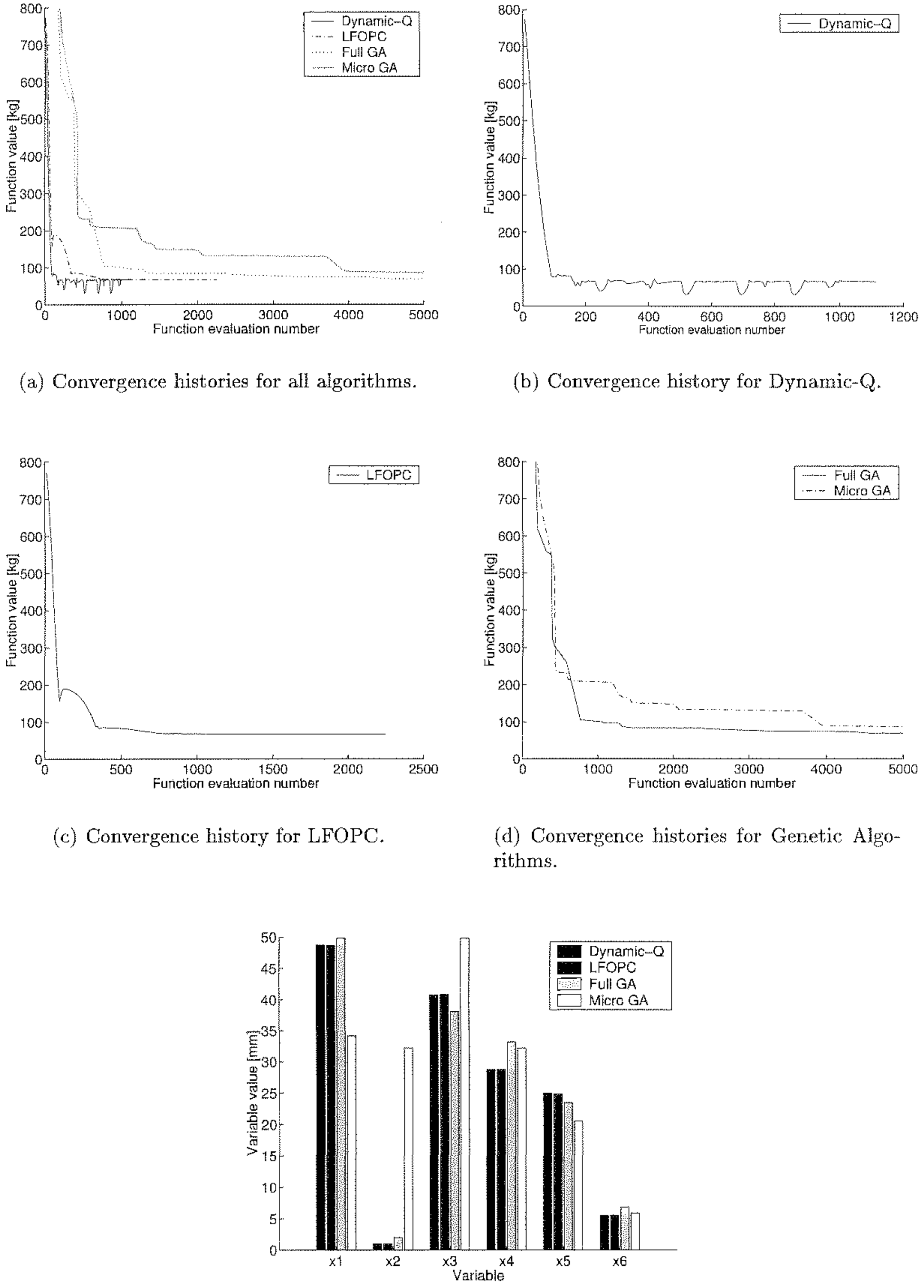
For this problem, both Dynamic-Q and LFOPC terminated at approximately the same design point, and therefore have very similar function values (around 66.9 kg). The point at which Dynamic-Q terminated, however, has a marginally lower working point displacement and is therefore a slightly better design. What is more, Dynamic-Q converged rapidly compared to the other algorithms. The micro GA in particular did not perform well for this problem, and terminated at a comparatively high assembly weight. The large feasible variable space explains why the two GA's do not perform well for this problem.

For an engineering application such as this, Dynamic-Q could have been stopped manually before the algorithm terminated due to its own criteria. For example, after 85 iterations (595 function evaluations) Dynamic-Q had already found a point ($\boldsymbol{x}^{85}$) with $F(\boldsymbol{x}^{85}) = 67.3$ kg and $\bar{g}(\boldsymbol{x}^{85}) = -2.74\ \mu$m. This manual termination is especially useful for the Dynamic-Q algorithm since it reaches the region of the solution extremely quickly. The non-monotonic and oscillatory convergence behavior of Dynamic-Q is discussed in the concluding discussion in Section 4.4.

It is clear from Figure H.1(b) that the displacement is not sensitive to x_2 (the stiffening guide diameter). This result indicates that the guides are not effective in preventing the deformation of the moving platform.

Figure H.1(f) shows that both the assembly weight and platform displacement are very sensitive to x_6 (the moving platform thickness). The high sensitivity to weight of this variable explains its low terminal value.

The above results also demonstrate that a platform, which is not required to be excessively stiff, can be made very maneuverable due to its low weight. This quality can be exploited in tasks where a large force is not required, such as in an assembly or placement process.



(a) Convergence histories for all algorithms.

(b) Convergence history for Dynamic-Q.

(c) Convergence history for LFOPC.

(d) Convergence histories for Genetic Algorithms.

(e) Terminal variable values.

Figure 4.5: Convergence histories and terminal variable values. (Original design with constraint set $\boldsymbol{g}(\boldsymbol{x})$ and $\bar{d} = 1000\mu\text{m}$).

4.3.2 Optimization of original design for constraint set $g(\boldsymbol{x})$ and $\bar{d} = 100$ μm.

	Dynamic-Q	LFOPC	Full GA	Micro GA
$F(\boldsymbol{x}^*)$ [kg]	213.511	213.517	246.311	341.617
$\bar{g}(\boldsymbol{x}^*)$ [μm]	-1.626e-03	-1.558e-03	-1.258e+01	-2.199e-01
Function evaluations	217	1925	2936	4004
x_1^* [mm]	86.796	86.760	88.886	84.980
x_2^* [mm]	1.000	1.000	1.977	8.812
x_3^* [mm]	76.741	76.607	84.004	125.994
x_4^* [mm]	55.157	55.628	74.239	65.450
x_5^* [mm]	44.214	43.873	22.483	43.967
x_6^* [mm]	12.716	12.772	14.671	8.812
Termination type	D$f$tol	L$x$tol	FG20	MG100

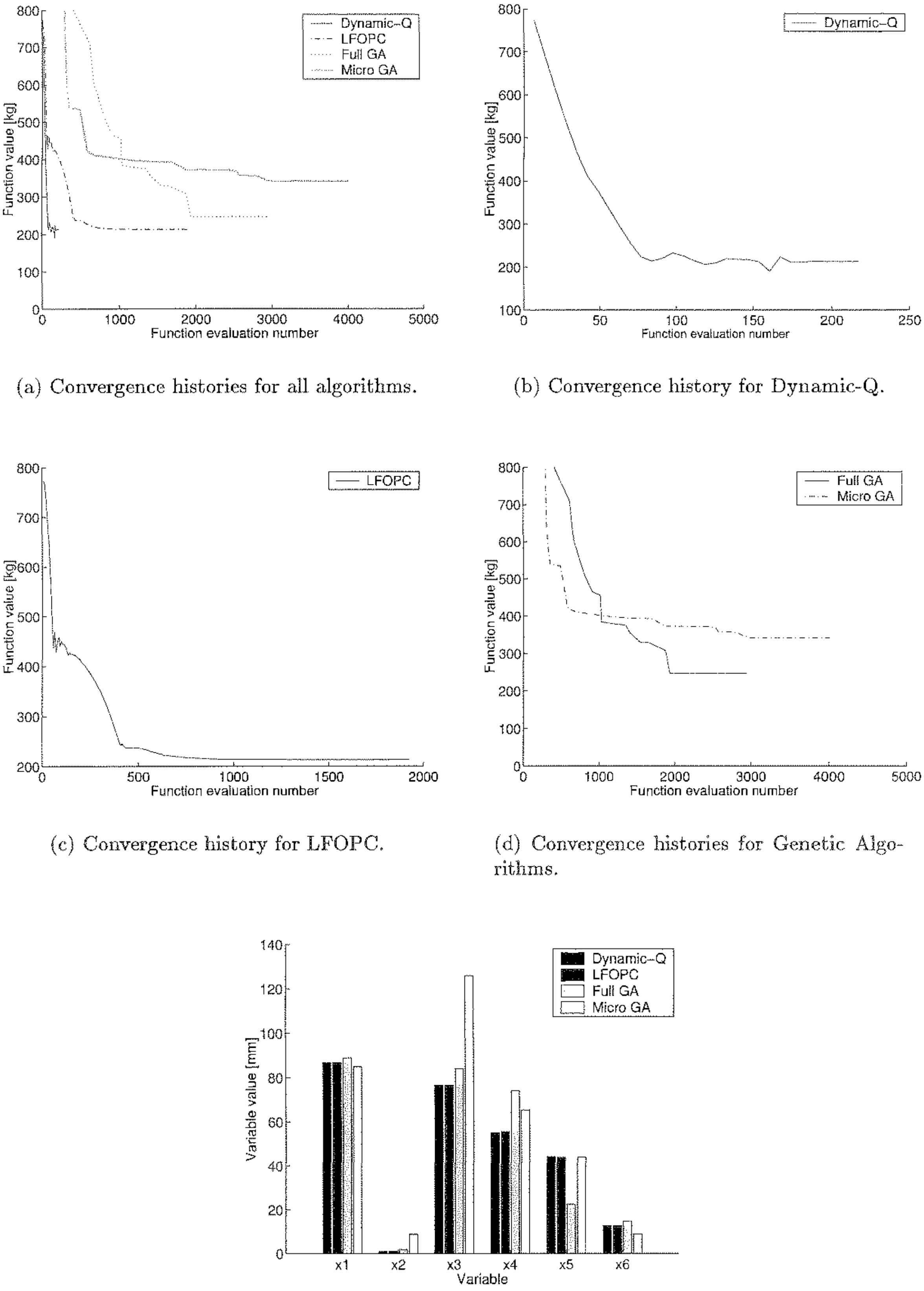
Table 4.3: Original design with constraint set $\boldsymbol{g}(\boldsymbol{x})$ and $\bar{d} = 100$ μm: Terminal values.

The results for this optimization run are tabulated in Table 4.3 and the convergence histories are shown in Figure 4.6.

Dynamic-Q and LFOPC terminated at very similar design points, corresponding to a function value of approximately 213.5 kg. As expected, this value is significantly higher than that for $\bar{d} = 1000$ μm. Again the micro GA performed badly, and terminated far from the best know design point (found by Dynamic-Q).

In this case, Dynamic-Q converged very quickly and did not oscillate very much in the region of the optimum as depicted in Figure 4.6(b). LFOPC again proved very robust and converges in a very regular and predictable fashion.

Again, the terminal values of x_2 and x_6 are relatively low. The function sensitivities are depicted in Figure H.2. The sensitivities to these two variables is similar to that observed in Section 4.3.1.



(a) Convergence histories for all algorithms.

(b) Convergence history for Dynamic-Q.

(c) Convergence history for LFOPC.

(d) Convergence histories for Genetic Algorithms.

(e) Terminal variable values.

Figure 4.6: Convergence histories and terminal variable values. (Original design with constraint set $\boldsymbol{g}(\boldsymbol{x})$ and $\bar{d} = 100\mu\text{m}$).

4.3.3 Optimization of original design for constraint set $g(x)$ and $\bar{d} = 10\ \mu$m.

	Dynamic-Q	LFOPC	Full GA	Micro GA
$F(\boldsymbol{x}^*)$ [kg]	839.990	840.567	1976.460	1232.120
$\bar{g}(\boldsymbol{x}^*)$ [μm]	1.529e-02	-1.270e-05	-7.895e-02	2.980e-02
Function evaluations	2107	1330	1602	3159
x_1^* [mm]	160.636	163.426	183.609	157.243
x_2^* [mm]	1.000	1.246	27.366	1.977
x_3^* [mm]	165.958	165.076	251.965	250.988
x_4^* [mm]	122.612	120.442	142.595	110.370
x_5^* [mm]	85.074	83.521	39.084	46.896
x_6^* [mm]	31.002	31.438	318.368	32.248
Termination type	Dmax	L$x$tol	FG20	MG100

Table 4.4: Original design with constraint set $\boldsymbol{g}(\boldsymbol{x})$ and $\bar{d} = 10\ \mu$m: Terminal values.

For this extreme accuracy requirement, the optimization results are as tabulated in Table 4.4, with convergence histories depicted in Figure 4.7.

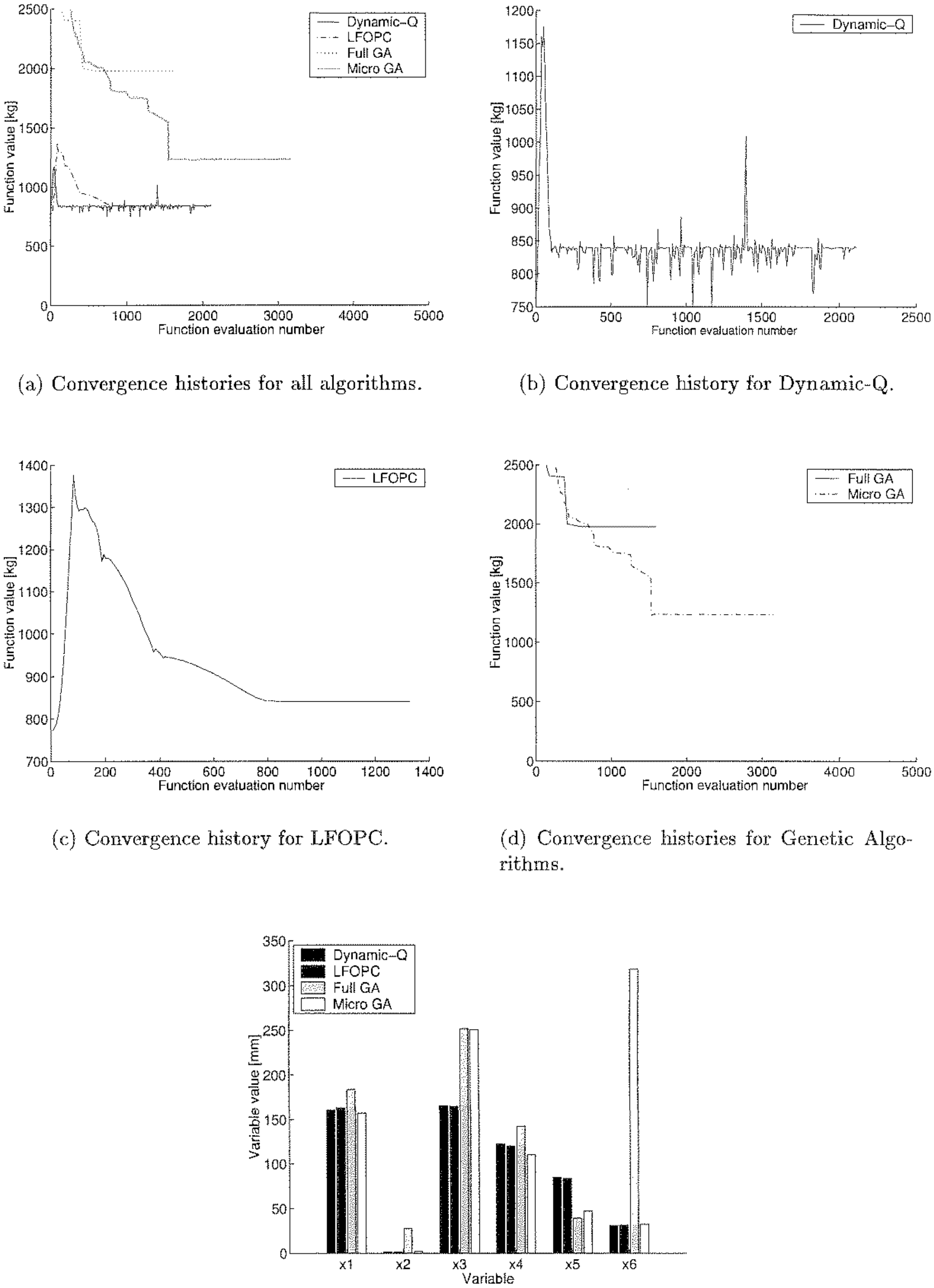
Again LFOPC and Dynamic-Q found similar optimum design points, with a corresponding weight of approximately 840 kg. However, the displacement constraint at the optimum design point determined by Dynamic-Q is slightly violated. Both GA's did very badly for this problem, again suggesting that the large design space hinders the performance of these algorithms.

As was the case for $\bar{d} = 1000\ \mu$m, Dynamic-Q oscillated for a long period of time in the region of the optimum before termination. This oscillation could be prevented by some manual user interaction. For example, after 43 iterations (301 function evaluations) a sufficient optimum value of $F(\boldsymbol{x}^{43}) = 839.9$ kg and $\boldsymbol{g}(\boldsymbol{x}^{43}) = 0.356 \times 10^{-1}\ \mu$m was found.

LFOPC could also have been manually stopped at an earlier point. For example, after 140 iterations (980 function evaluations) LFOPC had determined a design point with $F(\boldsymbol{x}^{140}) = 840.5$ kg and $\boldsymbol{g}(\boldsymbol{x}^{140}) = 0.49 \times 10^{-2}\ \mu$m.

There are several important physical issues which are apparent from these results. Firstly, the optimum assembly weight is very high (around 840 kg). This will limit the efficiency with which machining operations are carried out. Secondly, the high terminal variable values, for example the bottom transverse support strut diameter of 165 mm, leads to a design which will become difficult to manufacture and assemble. Thirdly, the low value of x_2 (1 mm, which corresponds to the lower bound for this variable) confirms that adding stiffening guides is not effective in adding stiffness to the design.

Figure H.3 confirms that both the objective and constraint function are relatively sensitive to all variables except x_2 at this optimum point determined by Dynamic-Q.



(a) Convergence histories for all algorithms.

(b) Convergence history for Dynamic-Q.

(c) Convergence history for LFOPC.

(d) Convergence histories for Genetic Algorithms.

(e) Terminal variable values.

Figure 4.7: Convergence histories and terminal variable values. (Original design with constraint set $\boldsymbol{g}(\boldsymbol{x})$ and $\bar{d} = 10\mu\text{m}$).

4.3.4 Optimization of original design for constraint set $\boldsymbol{G}(\boldsymbol{x})$ and $\bar{d} = 1000$ μm.

	Dynamic-Q	LFOPC	Full GA	Micro GA
$F(\boldsymbol{x}^*)$ [kg]	118.387	118.387	132.009	118.734
$\bar{g}(\boldsymbol{x}^*)$ [μm]	-3.931e-02	8.406e-03	-1.432e+02	-9.405e+00
Function evaluations	119	1267	1491	5000
x_1^* [mm]	47.157	47.230	45.029	47.573
x_2^* [mm]	10.000	10.001	20.959	10.000
x_3^* [mm]	40.553	40.482	45.480	40.411
x_4^* [mm]	30.000	30.004	31.566	30.000
x_5^* [mm]	30.000	30.000	35.049	30.039
x_6^* [mm]	30.000	30.000	30.235	30.039
Termination type	D$f$tol	L$x$tol	FG20	MGmax

Table 4.5: Original design with constraint set $\boldsymbol{G}(\boldsymbol{x})$ and $\bar{d} = 1000$ μm: Terminal values.

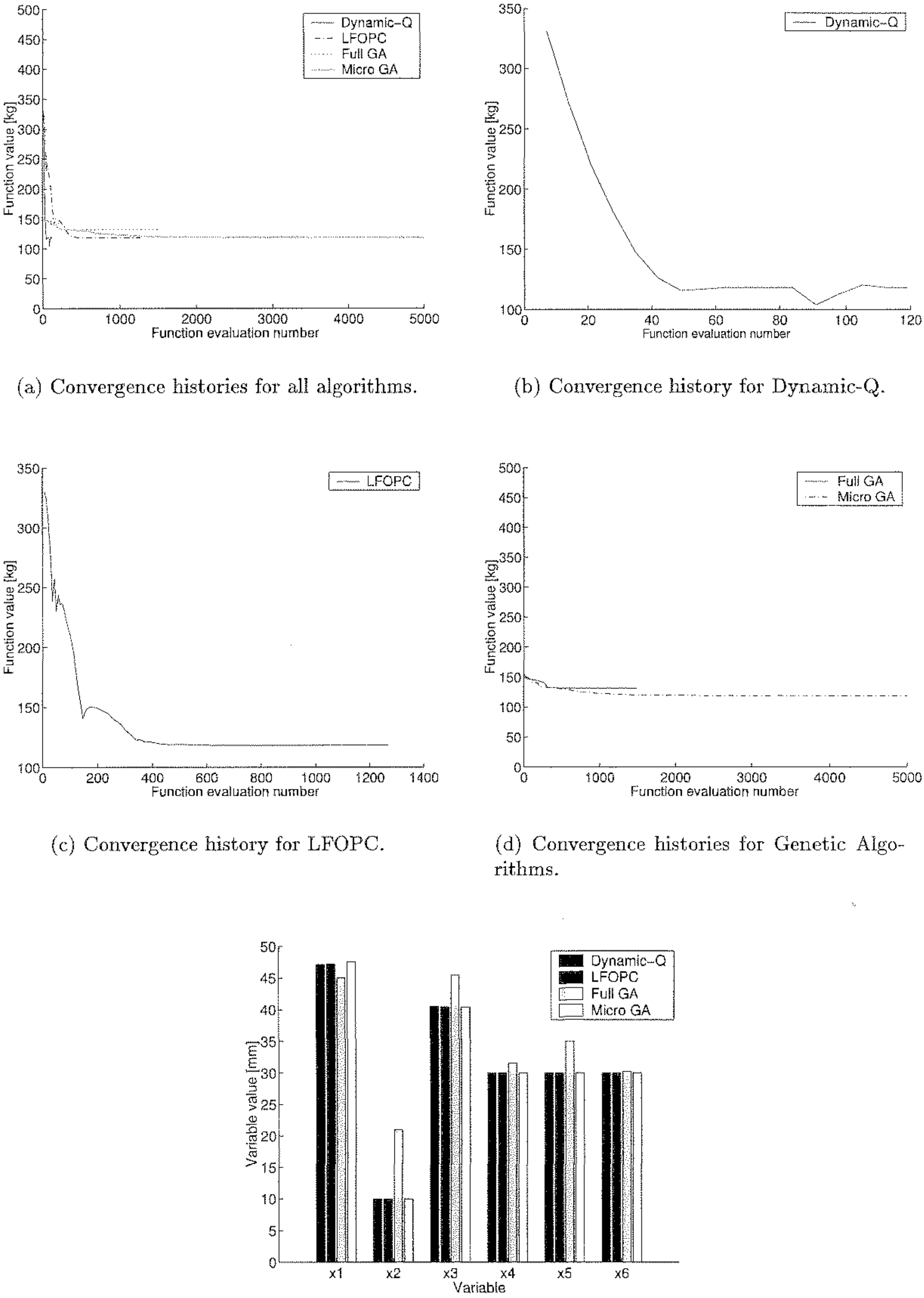
The results of this optimization run are presented in Table 4.5 with convergence histories in Figure 4.8.

Again, Dynamic-Q and LFOPC terminated at very similar solutions (with a weight of approximately 118.4 kg). The Dynamic-Q algorithm terminated on the function tolerance after very few oscillations. As expected, the more realistic variable lower bounds used in this problem result in a higher weight compared to the weight of only 66.9 kg found with the lenient set of side constraints (see Table 4.2).

In this case the micro GA found a very good solution, similar to the best known solution. This may be as a result of the smaller variable space (due to the "stricter" upper and lower variable bounds) in this problem. However, Dynamic-Q required only 119 function evaluations, compared to 1267 of LFOPC and 5000 of the micro GA.

The very flat convergence history of the micro GA suggests that the algorithm could have been terminated sooner (after around 2000 function evaluations). A function value of 119.8 kg and $\bar{g}(\boldsymbol{x}) = 1.897$ μm was found by this point. The very flat history in this case is due to the micro GA's ability to refine the solution (due to rebirth), finding slightly better solutions at regular intervals. This makes the stopping criterion of a given number of function evaluations with *no* improvement in function value a poor criterion. A minimum change in function value should be used in this criterion, similar to the tolerance used for the gradient-based algorithms.

It is also interesting to note that only x_1 (actuator diameter) and x_3 (bottom transverse support strut diameter) did not terminate at their lower bound for the best known solution (that of the Dynamic-Q algorithm). Figure H.4 confirms that the constraint function (which is active at the optimum design point) in particular, is very sensitivity to these variables.



(a) Convergence histories for all algorithms.

(b) Convergence history for Dynamic-Q.

(c) Convergence history for LFOPC.

(d) Convergence histories for Genetic Algorithms.

(e) Terminal variable values.

Figure 4.8: Convergence histories and terminal variable values. (Original design with constraint set $\boldsymbol{G}(\boldsymbol{x})$ and $\bar{d} = 1000\mu\text{m}$).

4.3.5 Optimization of original design for constraint set $\boldsymbol{G}(\boldsymbol{x})$ and $\bar{d} = 100\ \mu$m.

	Dynamic-Q	LFOPC	Full GA	Micro GA
$F(\boldsymbol{x}^*)$ [kg]	318.362	318.324	316.499	311.161
$\bar{g}(\boldsymbol{x}^*)$ [μm]	1.251e+01†	1.233e+01†	2.692e+01†	2.578e+01†
Function evaluations	2107	1554	3942	5000
x_1^* [mm]	61.739*	61.757*	59.902	60.000
x_2^* [mm]	52.104*	52.115*	50.000	50.000
x_3^* [mm]	100.256*	100.191*	98.904	100.000
x_4^* [mm]	50.589*	50.697*	49.922	50.000
x_5^* [mm]	50.113*	50.080*	49.922	49.961
x_6^* [mm]	29.929*	29.938*	34.364	30.196
Termination type	Dmax	Lxtol	FG20	MGmax

†-*extreme violation of constraint*

*-*slight violation on variable bound*

Table 4.6: Original design with constraint set $\boldsymbol{G}(\boldsymbol{x})$ aud $\bar{d} = 100\ \mu$m: Terminal values.

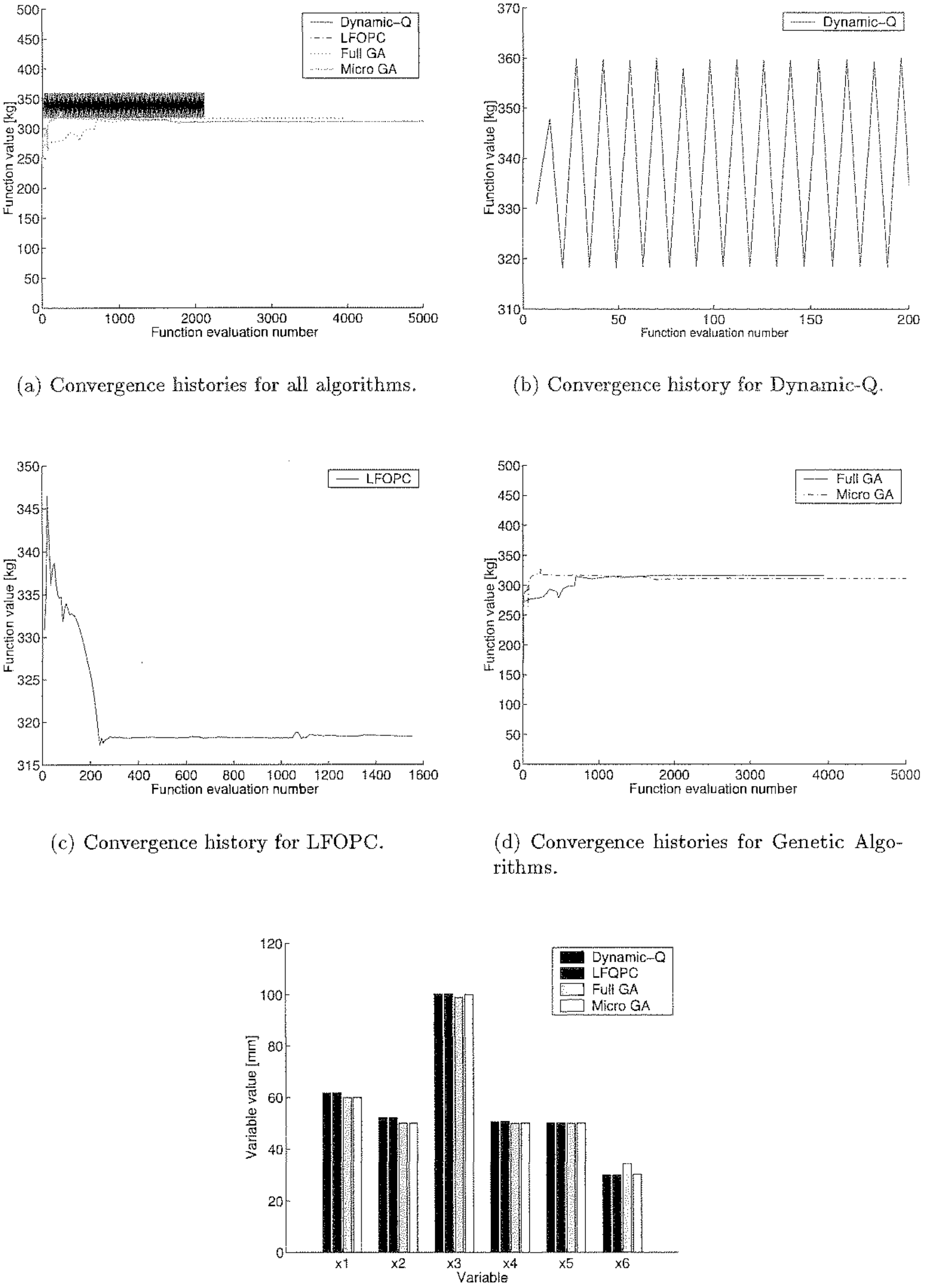
Table 4.6 contains the results for this optimization run, with corresponding convergence histories depicted in Figure 4.9.

The tabulated results show that no feasible design could be obtained for this specified value of $\bar{d}$. The "compromised optima" of the gradient-based methods and the "optima" of the GA's all result in extreme violations of the displacement constraint function.

Both Dynamic-Q and LFOPC violate the bounds on the variables slightly, and by doing so, greatly decrease the amount by which the displacement constraint is violated. The genetic algorithms are not able to compromise the solution in this way.

From Figure 4.9(b), it is clear that the behavior of the Dynamic-Q algorithm is very oscillatory, and should clearly have been stopped after a few oscillations. LFOPC proved robust, and the smooth convergence behavior allowed for termination on $\boldsymbol{x}$ tolerance.

The best (infeasible) solution lies at the point where the variables x_1 to x_5 are close to their upper bound and with x_6 close to its lower bound. At this design point, the displacement constraint is most sensitive to x_1, x_2 and x_4, which represent the actuator diameter, guide diameter, and upper transverse platform member diameter respectively.



(a) Convergence histories for all algorithms.

(b) Convergence history for Dynamic-Q.

(c) Convergence history for LFOPC.

(d) Convergence histories for Genetic Algorithms.

(e) Terminal variable values.

Figure 4.9: Convergence histories and terminal variable values. (Original design with constraint set $\boldsymbol{G}(\boldsymbol{x})$ and $\bar{d} = 100\mu\text{m}$).

4.4 Concluding discussion

From the optimization results presented, it is clear that the Dynamic-Q algorithm performs very well indeed for this problem. It consistently reaches the region of the solution very quickly. The oscillatory nature of the convergence history, however, suggests that some user interaction in terminating the optimization process would be advisable. This is especially true in the case of engineering problems where function evaluations are usually expensive, necessitating as few iterations as possible in reaching an acceptable optimal design, and an *exact* optimum is usually not necessary.

Figure 4.10 depicts the typical response of the penalty function around the displacement constraint boundary for this problem. The displacement function is know to be numerically noisy due to fact the displacements are only made available to four significant figures in the EDSAP output file. This noise, due to the displacement function, is further amplified by the penalty function parameter.

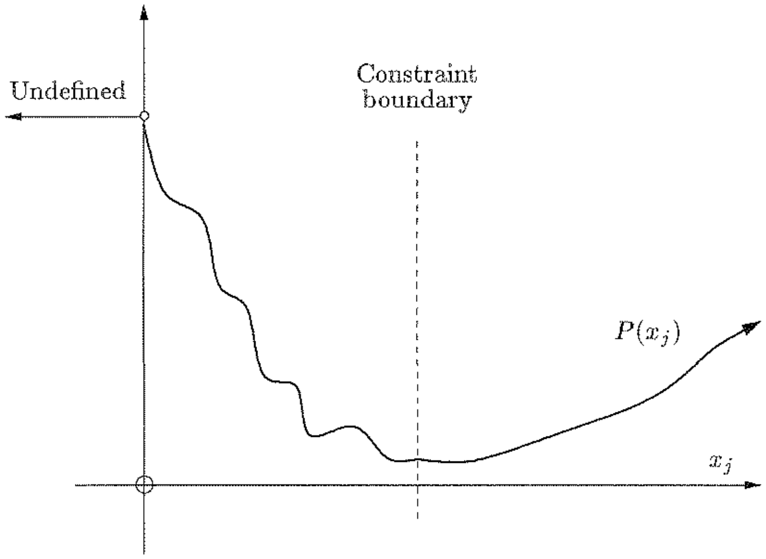


Figure 4.10: Illustration of noise on penalty function.

This explains the non-monotonic convergence behavior of Dynamic-Q, which is characteristic of the method, especially in the region of the active displacement constraint. The spikes correspond to deviations of the algorithm from the bed of the "optimum valley" as it progresses to the optimum. These deviations are always followed by correction steps. It is therefore recommended that the user monitors, and stops the algorithm interactively if a plateau is reached and constraints are sufficiently satisfied.

The LFOPC algorithm proved to be very robust, with very little oscillation. These two gradient-based methods terminated at what appear to be identical optima, and used in conjunction with each other, provide confidence in the computed optima. Overall, however, LFOPC requires more function evaluations for convergence than the Dynamic-Q algorithm.

The micro GA and the full GA are very evenly matched, and it is difficult to predict which algorithm will perform better for a particular application. In general the micro GA is able to refine the solution more efficiently once the region of the solution has been found for this problem. This is due to the fact that the rebirth process occurs often, generating new (sometimes better) genetic material. The full GA on the other hand relies on mutation which takes longer to refine a good design, unless the probability of mutation is set high. However, when the probability of mutation becomes too large, the algorithm tends to become little more than a random sampling algorithm.

This is in agreement with the "No Free Lunch" [53, 53] theorem which in simple terms states that if one tunes an algorithm to do well in one specific problem, in effect one is creating an algorithm that performs poorly for a large number of other different classes of problems.

The computed optimization results point to several practical design shortcomings. Firstly, it is evident from the results that the present design is not capable of meeting the 10 μm displacement constraint when realistic constraints are placed on the variable values. This manifests itself by the fact that, even for a relatively lenient displacement constraint, with $\bar{d} = 100$ μm, no feasible optima could be obtained. (See results in Table 4.6).

Secondly, if the side constraints were to be relaxed in order to attain higher accuracy, the weight would probably be restrictive. This is illustrated by the computed weight of 840 kg of the platform, when the 10 μm constraint is met, with the lenient set of bounds on the variables. Furthermore the converged variable values for this case, listed in Table 4.4, suggest that the stiffness is not at all sensitive to the size of the stiffening guides (x_2). The conclusion here is that the original design is not adequate and that an improved design should be sought. The design requires adjacent stiffening guides to be attached to each other in some way, in order for them to act as a unit and to prevent the rotation shown in Figure 3.8 (c).

Chapter 5

Optimization of an Improved Concept Design

In the previous chapter it was shown that the proposed original concept design is not feasible, and will not be able to carry out machining tasks to the required 10 μm accuracy. For this reason design changes are to be implemented in order to carry out machining tasks to the required machining tolerance.

In this chapter an improved concept design, based on a design suggested by Du Plessis [61], is proposed and analyzed using FEM. The design modifications are intended to increase the stiffness of the platform. Again, the dimensional design is to be optimized with respect to weight. Here an additional stress constraint is added in order to ensure that the design is structurally capable of supporting the loads experienced during machining operations. Finally, stiffness maps are drawn up for the design and the importance of toolpath planning in the workspace is briefly discussed.

5.1 The improved planar platform design

In the previous chapter it was concluded that a possible reason for the failure of the original concept design is the lack of interaction between adjacent stiffening guides. The best way to overcome this problem is to add a thin plate between adjacent guide rails, thus greatly adding to the inertia of the structure.

Design modifications are necessary to accommodate the plates and thus avoid interference between the stiffening plates and the actuators. For this reason an offset is introduced between the actuator and the plate itself as shown in Figure 5.1. The two center stiffening guides are considered redundant and are removed from the assembly.

Figure 5.1 shows a schematic representation of the modified design. The differences between the current design and the existing concept design are clearly illustrated. Again, the moving platform slab has not been shown for clarity.

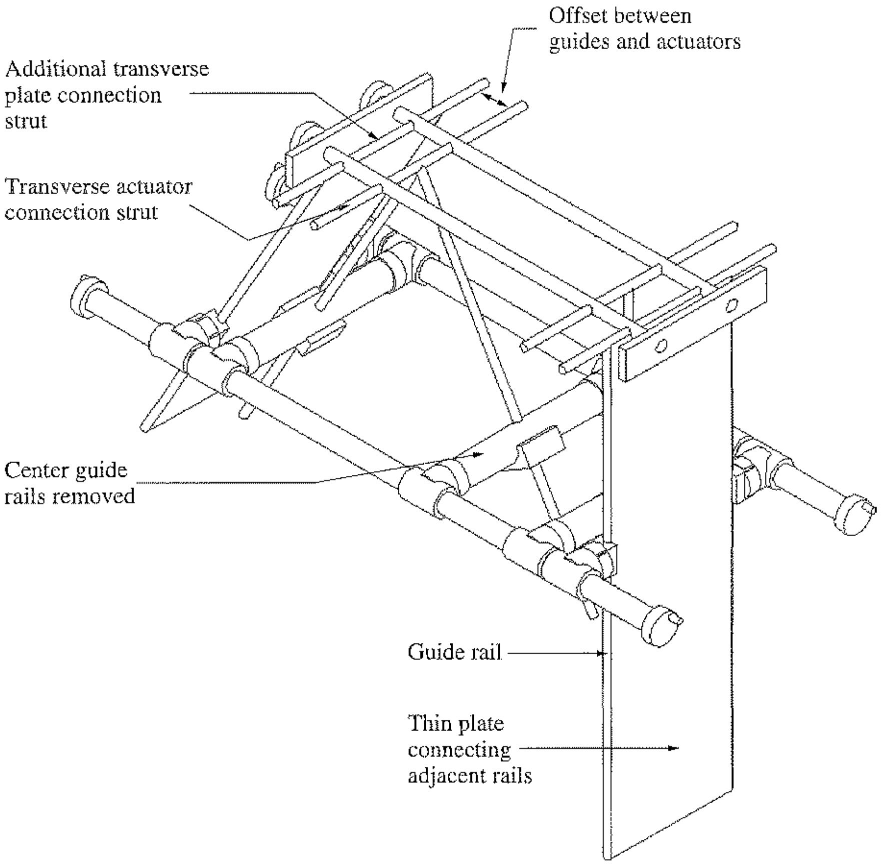

Offset between guides and actuators
Additional transverse plate connection strut
Transverse actuator connection strut
Center guide rails removed
Guide rail
Thin plate connecting adjacent rails

Figure 5.1: Schematic of improved planar platform design.

5.2 Input file and mesh generation

The finite element analysis of the improved platform design is, again, carried out with EDSAP. The (EDSAP) input file generating subroutine which was coded for this platform geometry, creates a mesh in such a way that the discretization is regular and undistorted. With reference to Figure 5.2 the functioning of the mesh generator is detailed below:

1. Specify the number of (platform) increments (n_1) that are desired along l_1.

2. Calculate the length of the elements along l_1, denoted by δl_1.

3. Calculate the number of platform increments, n_2, along l_2 that will come closest to having $\delta l_2 = \delta l_1$.

4. Calculate δl_2 using $\delta l_2 = \frac{l_2}{n_2}$.

5. Similarly, calculate δl_3, δl_4 and δl_5.

Therefore by specifying only n_1 the user can control the size of the elements in the mesh. The section on the moving platform bounded by l_1 and l_3, for example, will have shell elements of size $\delta l_1 \times \delta l_3$.

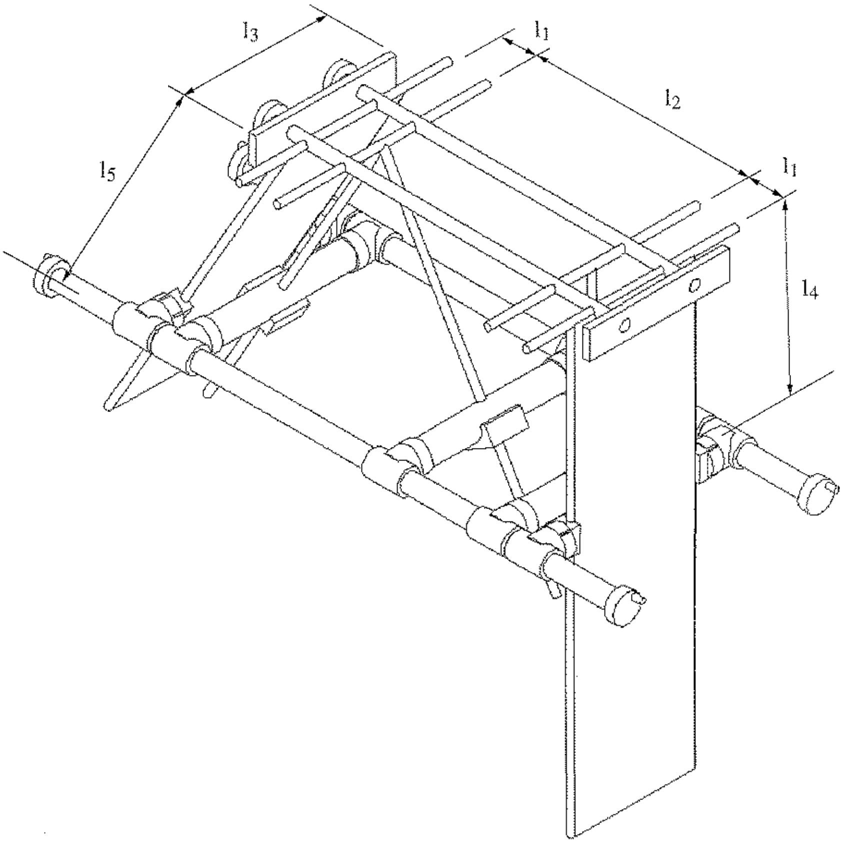


Figure 5.2: Illustration of parameters used for mesh generation.

5.3 Optimization of the improved design

In this section the optimization methodology is applied to the improved concept design. The relevant definitions of the design variables, the constraint functions as well as the cost function now follow.

5.3.1 Design variables

For consistency, the platform is fixed in a configuration similar to that used for the original concept design in Chapter 4. For this design, nine design variables are used. They are

- x_1 - moving platform thickness,

- x_2 - left hand side plate thickness,
- x_3 - right hand side plate thickness,
- x_4 - actuator diameter,
- x_5 - guide diameter,
- x_6 - transverse plate connecting strut diameter,
- x_7 - transverse actuator connecting strut diameter,
- x_8 - lateral platform member diameter, and
- x_9 - main transverse supporting strut diameter.

Again, the maximum number of design variables are used, without doing sensitivity analyses in order to eliminate those variables to which the objective function is not sensitive. This allows for a better test of the various optimization methods.

5.3.2 Constraints

From the optimization carried out in Chapter 4, realistic upper and lower bounds are known for the various design variables. Given this information, the constraint set that is placed on the the design variables is detailed below.

The upper and lower bounds on the variables can again be written in the form:

$$\hat{k}_i \leq x_i \leq \check{k}_i, \qquad i = 1, 2, \ldots, 9 \tag{5.1}$$

where the numerical values for the lower and upper bounds are taken as

$$\begin{aligned} \hat{\boldsymbol{k}} &= [30 \;\; 1 \;\; 1 \;\; 10 \;\; 10 \;\; 30 \;\; 30 \;\; 30 \;\; 30]^T \\ \check{\boldsymbol{k}} &= [40 \;\; 30 \;\; 30 \;\; 60 \;\; 50 \;\; 50 \;\; 50 \;\; 50 \;\; 100]^T \end{aligned} \tag{5.2}$$

The corresponding constraint functions are given by

$$\begin{aligned} \hat{g}_i &= \hat{k}_i - x_i, \qquad i = 1, 2, \ldots, 9 \\ \check{g}_i &= x_i - \check{k}_i, \qquad i = 1, 2, \ldots, 9 \end{aligned} \tag{5.3}$$

The constraint on the displacement of the working point is again written in the form:

$$\bar{g}_1(\boldsymbol{x}) = d(\boldsymbol{x}) - \bar{d} \leq 0 \tag{5.4}$$

In Chapter 4, no attention was paid to whether or not the assembly is structurally capable of carrying the applied load. In order to be sure that the design will not fail structurally, an additional constraint is placed on the stress that is experienced by each element in the assembly. Formally this stress constraint is

$$\bar{g}_2(\boldsymbol{x}) = \max\{\sigma_{VMi}(\boldsymbol{x}) - \frac{\sigma_{yi}}{SF} \quad , \quad i = 1, 2, \ldots, m\} \leq 0 \tag{5.5}$$

where $\sigma_{VMi}(\boldsymbol{x})$ is the von Mises stress of element i, σ_{yi} is the yield stress of element i, SF is a factor of safety with $SF = 2$ and m is the total number of elements in the model.

These constraints may now be represented in vector form by

$$\boldsymbol{g}(\boldsymbol{x}) = [\hat{\boldsymbol{g}}(\boldsymbol{x}) \quad \check{\boldsymbol{g}}(\boldsymbol{x}) \quad \bar{\boldsymbol{g}}(\boldsymbol{x})]^T \leq \boldsymbol{0} \tag{5.6}$$

5.3.3 Cost function

The cost function that is used for this optimization is consistent with that used in Chapter 4. The structure is minimized with respect to weight subject to the constraints given by (5.6). The explicit objective function is therefore:

$$F(\boldsymbol{x}) = \sum_{i=1}^{m}(w_i) \tag{5.7}$$

where w_i represents the weight of finite element (i) and m represents the total number of elements in the model.

Again, a penalty function approach is used to handle the constraints.

5.4 Analysis of optimization results

The optimization is carried out for three different and progressively smaller values of $\bar{d}$, where $\bar{d}$ is the maximum allowable working point displacement. Because of the robustness of the gradient-based methods used in this study, the infeasible starting $\boldsymbol{x}$ vector $\boldsymbol{x}^0 = [0 \; 0 \; 0 \; 0 \; 0 \; 0 \; 0 \; 0 \; 0]^T$ may be used. Furthermore, it was noted in Section 3.5.1 that the lower transverse supporting struts need to be supported, in order to prevent these critical members from bending. For this study it is assumed that, for each supporting strut, the design allows for supports 100 mm from the actuator connection point (on either side).

5.4.1 Optimization of improved design for constraint set $g(x)$ and $\bar{d} = 1000$ μm.

	Dynamic-Q	LFOPC	Full GA	Micro GA
$F(\boldsymbol{x}^*)$ [kg]	115.181	115.179	117.938	115.337
$\bar{g}_1(\boldsymbol{x}^*)$ [μm]	-7.792e+02	-7.792e+02	-7.906e+02	-7.822e+02
$\bar{g}_2(\boldsymbol{x}^*)$ [MPa]	-7.350e+01	-7.350e+01	-7.367e+01	-7.367e+01
Function evaluations	50	1540	5000	4837
x_1^* [mm]	30.000	30.000	30.039	30.020
x_2^* [mm]	1.000	1.000	1.170	1.000
x_3^* [mm]	1.000	1.000	1.000	1.000
x_4^* [mm]	10.000	10.000	10.196	10.098
x_5^* [mm]	10.000	10.000	10.626	10.235
x_6^* [mm]	30.000	30.000	30.078	30.039
x_7^* [mm]	30.000	30.000	32.583	30.000
x_8^* [mm]	30.000	30.000	31.135	30.000
x_9^* [mm]	30.000	30.000	31.370	30.000
Termination type	D$x$tol	L$x$tol	FGmax	MG100

Table 5.1: Improved design with $\bar{d} = 1000$ μm: Terminal values.

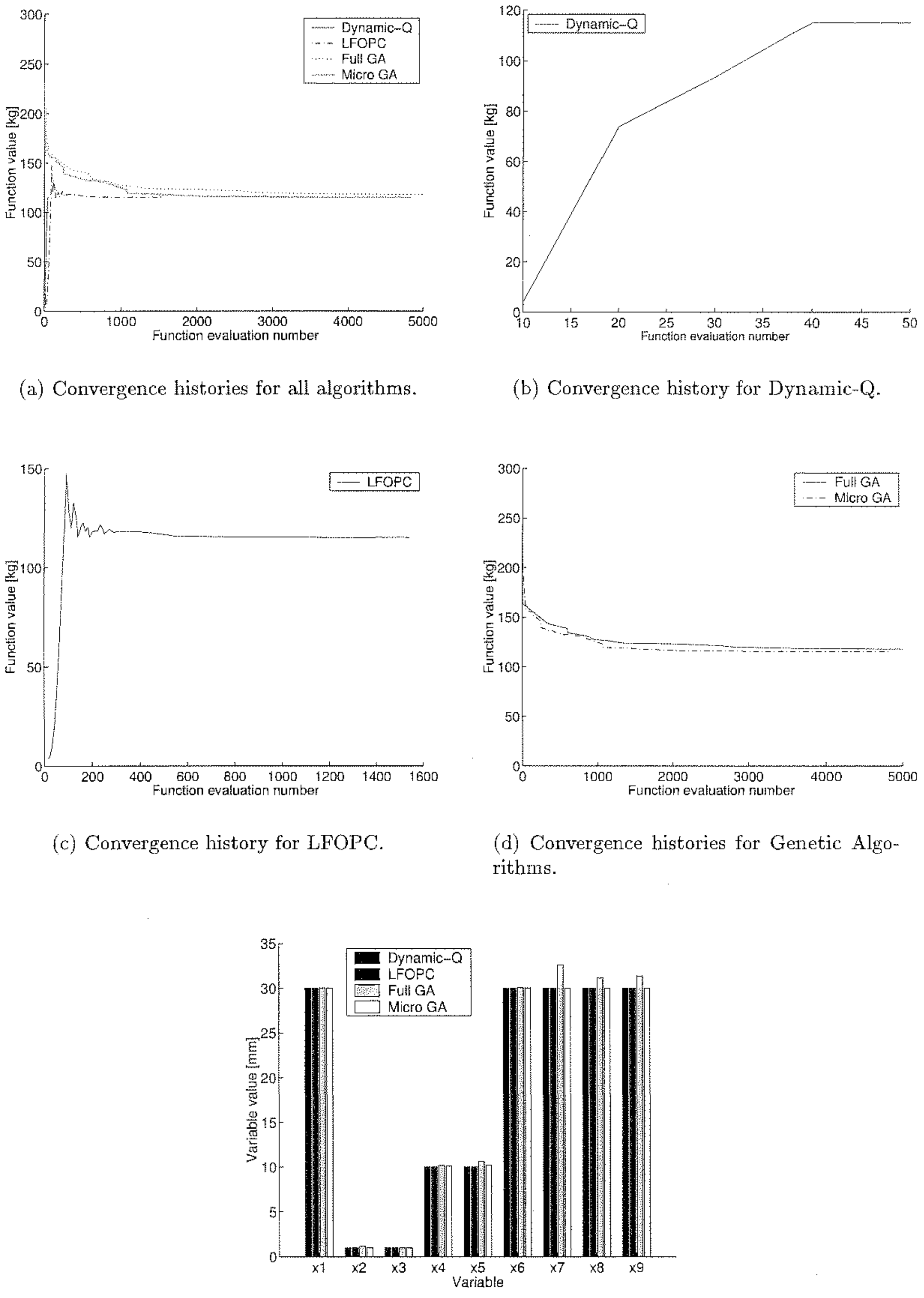
The optimization results for this case are tabulated in Table 5.1, with convergence histories depicted in Figure 5.3.

For this problem all the algorithms found similar optimum design points. Although LFOPC found a *slightly* better function value, Dynamic-Q converged in only 5 iterations (50 function evaluations). This indicates that Dynamic-Q performs well when the (noisy) displacement and stress constraints, $\bar{g}_1(\boldsymbol{x}) \leq 0$ and $\bar{g}_2(\boldsymbol{x}) \leq 0$, are not active. Clearly, even though the gradient-based methods started at the infeasible point $\boldsymbol{x}^0 = [0, 0, 0, \dots, 0]^T$, it did not hamper the performances of these algorithms. Again, the flat convergence histories of the GA's suggest that their stopping criteria are not effective.

All variables stopped at (or close to) their lower bound, and only these lower bound constraints are active at the optimum design point. That is, neither the stress nor the displacement constraints are active. The computed design therefore easily complies with the prescribed accuracy requirement.

Figures H.6 and H.7 depict the sensitivity of the objective and constraint functions to the variable values around the best known dimensional design. It is apparent that the displacement of the working point is very sensitive to the variable x_3 (the left hand side plate thickness). This indicates that the parameters related to the improved design (i.e. the additional stiffening plates) have a large influence on the working point displacement.

The stress sensitivity to x_2 has a "kink" around $x_2 = 1.3$ mm, as depicted in Figure H.6(b). The stress constraint function is a function of the maximum stress out of all the elements. This kink occurs at a point where the maximum stress in the system changes from occurring at one element to another.



(a) Convergence histories for all algorithms.

(b) Convergence history for Dynamic-Q.

(c) Convergence history for LFOPC.

(d) Convergence histories for Genetic Algorithms.

(e) Terminal variable values.

Figure 5.3: Convergence histories and terminal variable values (Improved design with $\bar{d} = 1000\mu\text{m}$).

5.4.2 Optimization of improved design for constraint set $g(x)$ and $\bar{d} = 100$ μm.

	Dynamic-Q	LFOPC	Full GA	Micro GA
$F(\boldsymbol{x}^*)$ [kg]	117.964	117.984	132.215	119.340
$\bar{g}_1(\boldsymbol{x}^*)$ [μm]	-2.912e-03	-1.894e-01	-3.626e+01	-1.136e+01
$\bar{g}_2(\boldsymbol{x}^*)$ [MPa]	-8.000e+01	-8.004e+01	-8.657e+01	-8.102e+01
Function evaluations	780	2470	3719	4317
x_1^* [mm]	30.000	30.000	30.000	30.000
x_2^* [mm]	1.000	1.001	1.227	1.000
x_3^* [mm]	1.000	1.001	1.624	1.000
x_4^* [mm]	18.938	19.030	26.438	22.524
x_5^* [mm]	10.000	10.000	14.462	10.000
x_6^* [mm]	30.000	30.000	41.742	30.039
x_7^* [mm]	30.000	30.000	30.274	30.000
x_8^* [mm]	30.000	30.000	30.078	30.000
x_9^* [mm]	30.672	30.582	33.699	30.000
Termination type	D$x$tol	L$x$tol	FG20	MG100

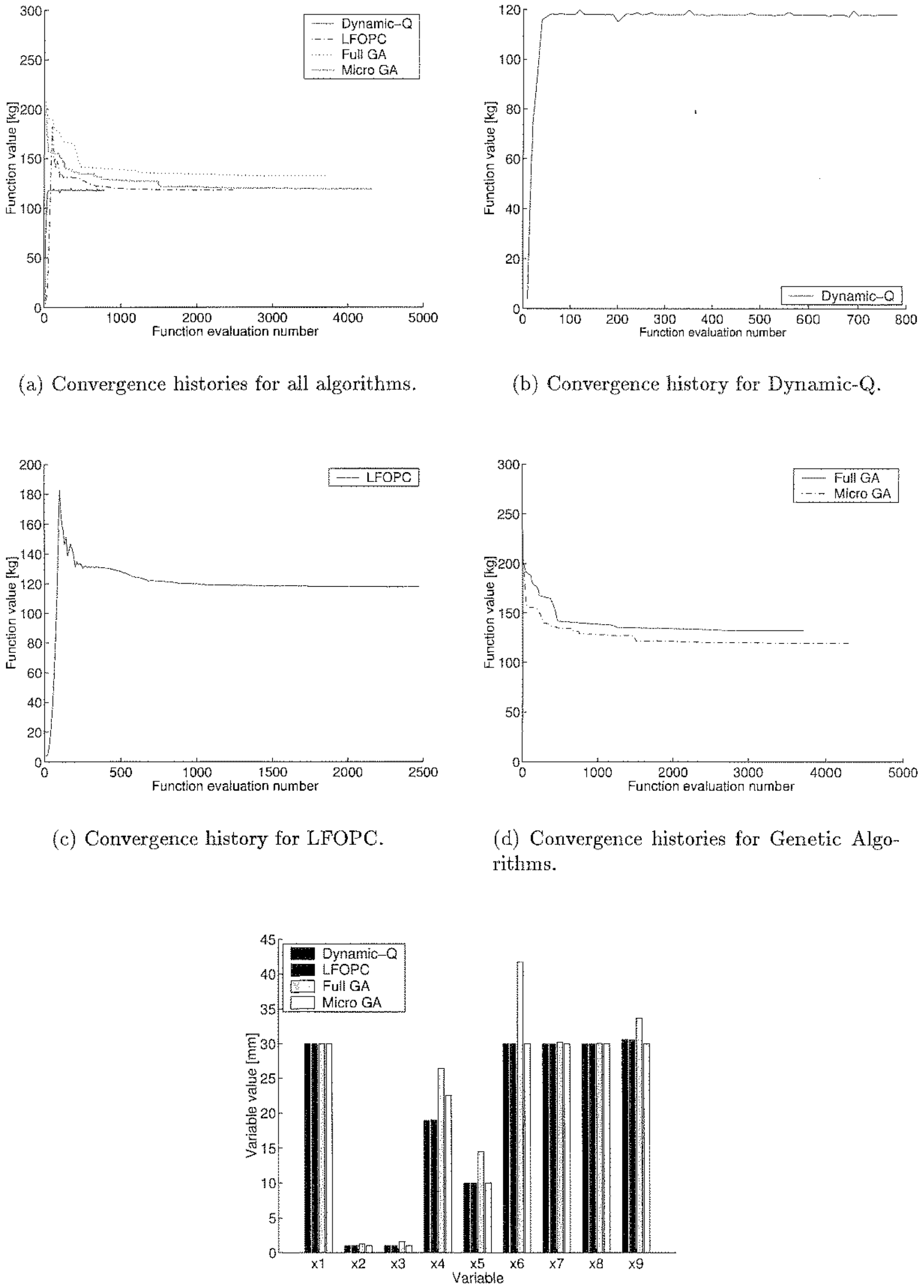
Table 5.2: Improved design with $\bar{d} = 100$ μm: Terminal values.

The optimization results for this case are listed in Table 5.2 and the convergence histories are depicted in Figure 5.4.

For this problem only the full GA did not find a good design. As with the previous problem, Dynamic-Q found the region of the solution quickly, but in this case took some time to terminate. Again, Dynamic-Q could have stopped far sooner, suggesting again that user interaction would be beneficial. For example after 30 iterations (300 function evaluations) Dynamic-Q found a point with $F(\boldsymbol{x}^{30}) = 117.97$ kg, $\bar{g}_1 = 0.182399 \times 10^{-2}$ μm and $\bar{g}_2 = -0.799988 \times 10^2$ MPa.

Of all the variables, only the optimum value for x_4 (the actuator diameter) does not lie on or near a lower bound. A reason for this is that the assembly, weight is not very sensitive to x_4, whereas the displacement constraint (which is the only active constraint other than the variable bounds) is moderately sensitive to this variable. This is depicted in Figures H.8(d).

From the variable values, depicted in Figure 5.4(e), it is apparent that neither GA had fully converged at the time of termination. This is clear from the fact that most of the variable values are in the vicinity of the best known design point.



(a) Convergence histories for all algorithms.

(b) Convergence history for Dynamic-Q.

(c) Convergence history for LFOPC.

(d) Convergence histories for Genetic Algorithms.

(e) Terminal variable values.

Figure 5.4: Convergence histories and terminal variable values (Improved design with $\bar{d} = 100\mu\text{m}$).

5.4.3 Optimization of improved design for constraint set $g(\boldsymbol{x})$ and $\bar{d} = 10\ \mu$m.

	Dynamic-Q	LFOPC	Full GA	Micro GA
$F(\boldsymbol{x}^*)$ [kg]	285.563	302.708	301.473	294.501
$\bar{g}_1(\boldsymbol{x}^*)$ [μm]	2.098e-01†	2.098e-01†	4.295e-01†	4.262e-01†
$\bar{g}_2(\boldsymbol{x}^*)$ [MPa]	-1.207e+02	-1.206e+02	-1.206e+02	-1.206e+02
Function evaluations	3010	4350	3584	3739
x_1^* [mm]	39.902	38.113	37.593	37.084
x_2^* [mm]	13.081	17.552	17.798	17.004
x_3^* [mm]	12.867	15.773	17.515	15.926
x_4^* [mm]	61.466*	60.964*	60.000	60.000
x_5^* [mm]	10.000	18.269	11.174	12.661
x_6^* [mm]	30.129	31.357	30.078	30.000
x_7^* [mm]	50.038*	50.081*	49.922	49.961
x_8^* [mm]	29.945	29.957	30.117	30.000
x_9^* [mm]	99.939	100.065*	100.000	100.000
Termination type	Dmax	Lmax	FG20	MG100

†*-slight violation of constraint*

**-slight violation on variable bound*

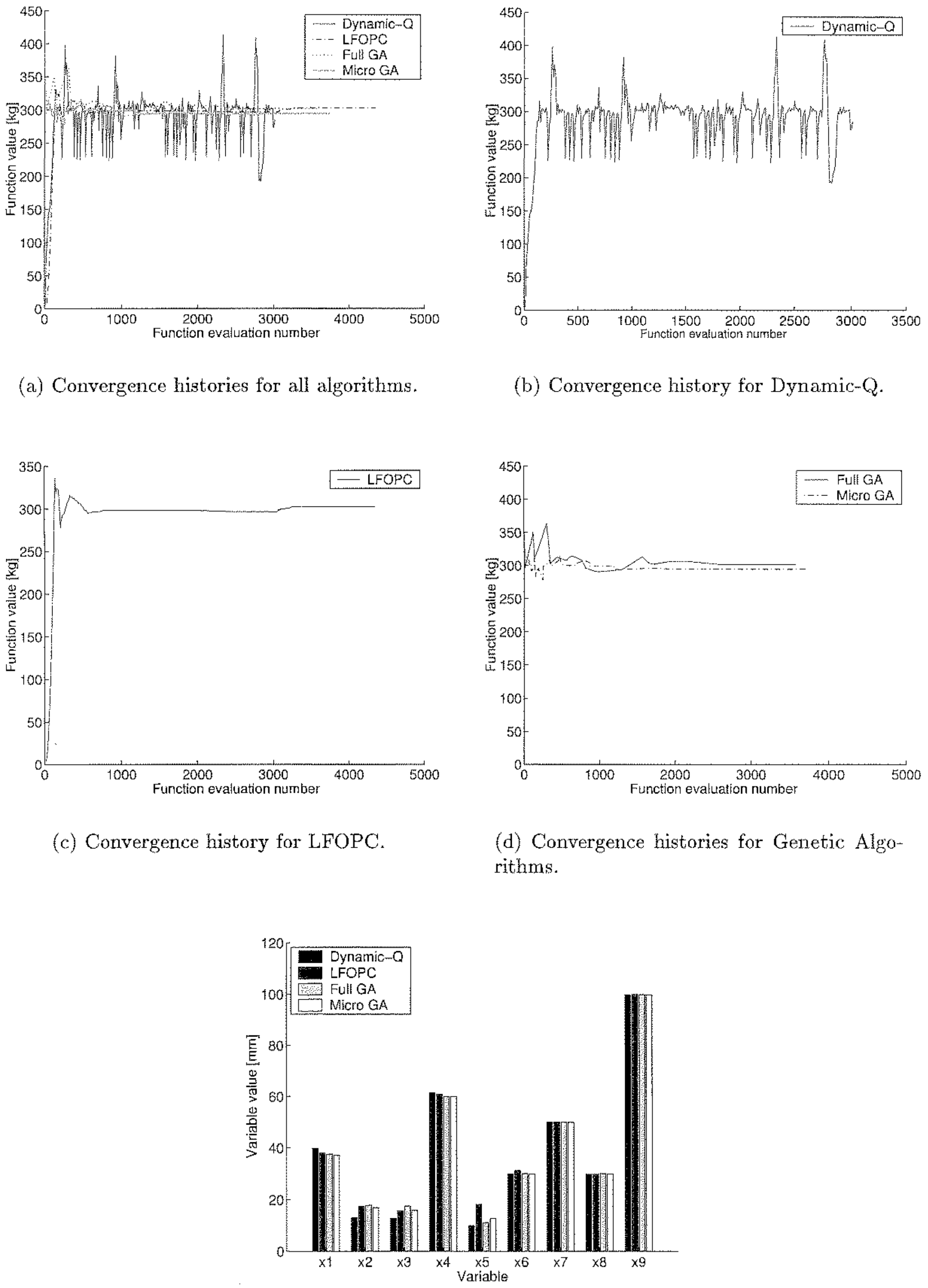
Table 5.3: Improved design with $\bar{d} = 10\ \mu$m: Terminal values.

The results for this final run with the most stringent deflection specification are tabulated in Table 5.3 with convergence histories depicted in Figure 5.5.

For this case Dynamic-Q found a solution that is significantly better than that of the other algorithms. The Dynamic-Q solution is approximately 10 kg lighter than the next best solution. The optimal weight (286 kg) is relatively low, considering the high stiffness of the design.

It is noticed that the solutions of the gradient-based methods are slightly compromised, in that the displacement constraint as well as the upper bound of some of the variables, namely x_4 and x_7, are slightly violated. This ability to compromise variable values is not available to the GA's since a fixed upper and lower bound is specified. This design compromisation explains why LFOPC and Dynamic-Q did not converge sharply, but terminate due the algorithms exceeding the maximum allowable number of steps. However, the compromised solution does indeed correspond to an acceptable design with an effective $\bar{d} \approx 10.2\ \mu$m.

From the convergence history of the LFOPC algorithm (Figure 5.5(c)), an increase in the function value is noted after approximately 3000 function evaluations (300 iterations). This coincides with the end of phase 0 of the algorithm. At this point LFOPC had found a design point with $F(\boldsymbol{x}^{300}) = 296.7$ kg and $\bar{g}_1(\boldsymbol{x}^{300}) = 0.217\ \mu$m. After this, the penalty parameter is increased, and a final optimum design point is determined.



(a) Convergence histories for all algorithms.

(b) Convergence history for Dynamic-Q.

(c) Convergence history for LFOPC.

(d) Convergence histories for Genetic Algorithms.

(e) Terminal variable values.

Figure 5.5: Convergence histories and terminal variable values (Improved design with $\bar{d}$ = 10μm).

5.4.4 Deformation analysis of the improved planar platform design

The best dimensional design (given in Table 5.3) for the case where $\bar{d} = 10$ μm was prescribed, is now used to analyze the displacement characteristics of the improved design. From Table 5.3 it is clear that Dynamic-Q found the best dimensional design with respect to weight, albeit slightly infeasible ($\bar{g}_1(\boldsymbol{x}^*) = 0.2098$ μm). The deflection of the platform is calculated under load and depicted in Figure 5.6. For clarity the displacement is amplified 2000 times in this figure.

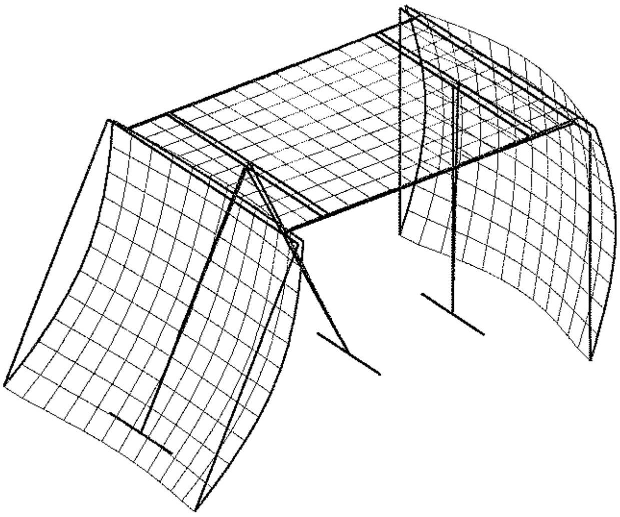


Figure 5.6: Deformation of the improved design under load.

From the results of the finite element analysis, it is clear that there is far less out of plane displacement than in the original design. The deformation in the X direction has been significantly reduced from that experienced by the originally proposed design. What is more, the undesired rotation is almost completely eliminated.

5.5 Stiffness mapping of the improved planar platform design

Gosselin [9] suggests the use of stiffness or conditioning maps in order to predict singularities in the workspace. These maps reveal zones where the stiffness is not acceptable and also help the designer by providing a more accurate representation of the manipulator properties.

The derivation of the stiffness of a planar parallel platform using the regular Jacobian method is presented in Appendix A. In this section the approach taken by Clinton *et al.* [10] is followed in which the stiffness is calculated using the finite element method.

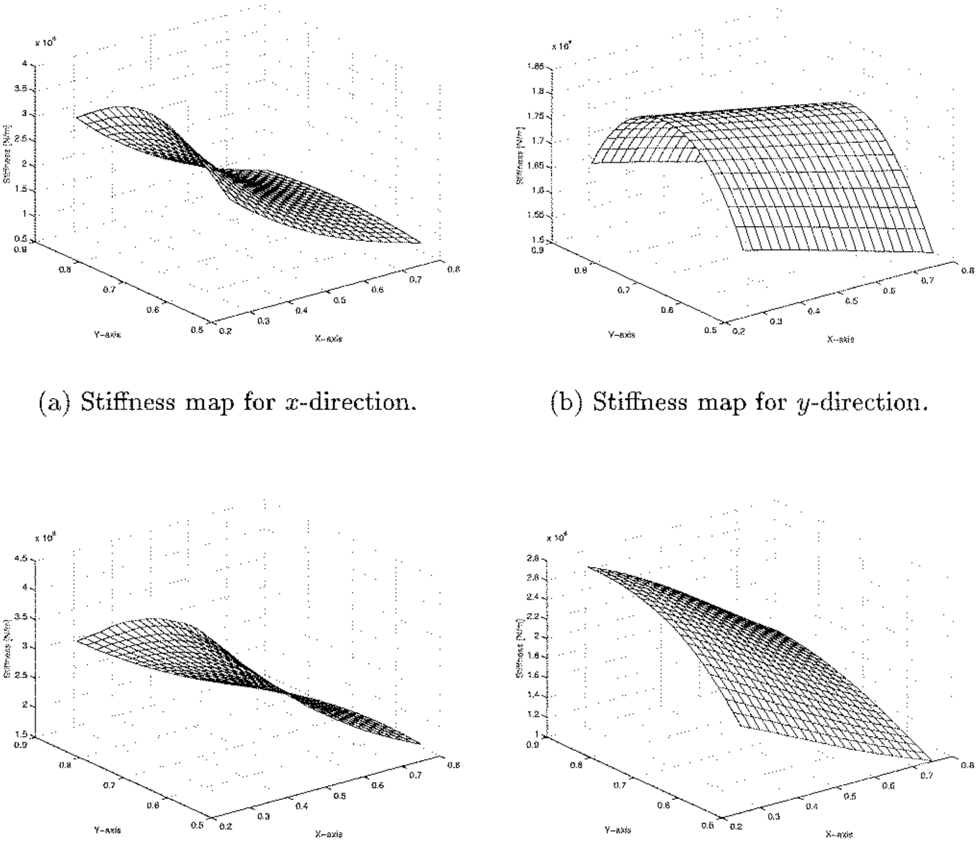
In this study the platform orientation is held at $\phi = 0°$ and the working point is moved to different points in the workspace. At each point the engineering (or single dimensional) stiffness is calculated by dividing the magnitude of the applied force in a certain direction with the displacement in the same direction.

Real stiffness in this study is calculated by dividing the magnitude of a force vector $\boldsymbol{f} = 1000\hat{\imath} + 1000\hat{\jmath} + 1000\hat{k}$ N by the *total magnitude* of the resultant working point displacement. This resultant displacement is not necessarily in the same direction as the applied force as is the case for the single dimensional stiffness.

From the stiffness maps depicted in Figure 5.7 it can be seen that the magnitude of the y stiffness is noticeably lower than the x and z stiffnesses. The high z stiffness suggests that the out of plane displacements are effectively prevented.

The complex shapes of these stiffness maps suggest that the optimization process should be carried out with the moving platform in more than one posture. A better strategy would be to perform the optimization over a toolpath, or even over the entire workspace.

Section 5.6 shows that these maps can be used, to a certain extent, to predict the existence of singularities or ill-conditioned postures in the workspace. These ill-conditioned regions represent zones with unacceptably low stiffness.



(a) Stiffness map for x-direction.

(b) Stiffness map for y-direction.

(c) Stiffness map for z-direction.

(d) Stiffness map for real stiffness.

Figure 5.7: Stiffness maps for optimal weight design and $\phi = 0°$.

5.6 Singularity analysis

Figure 5.8 depicts a planar parallel platform in a singular configuration. In such a posture the Jacobian matrix becomes rank deficient and the platform becomes uncontrollable. For the geometry used in this study, the singularity occurs at $(x_p, y_p) = (0.4268, 0.3232)$ and $\phi = -45°$.

The stiffness of the platform in this singular position is theoretically zero. In order to test the validity of the finite element model used in this study, the platform is analyzed in the vicinity of such a configuration. The working point is held at the x and y coordinates given above and rotated through the singular configuration, and the corresponding stiffness is monitored.

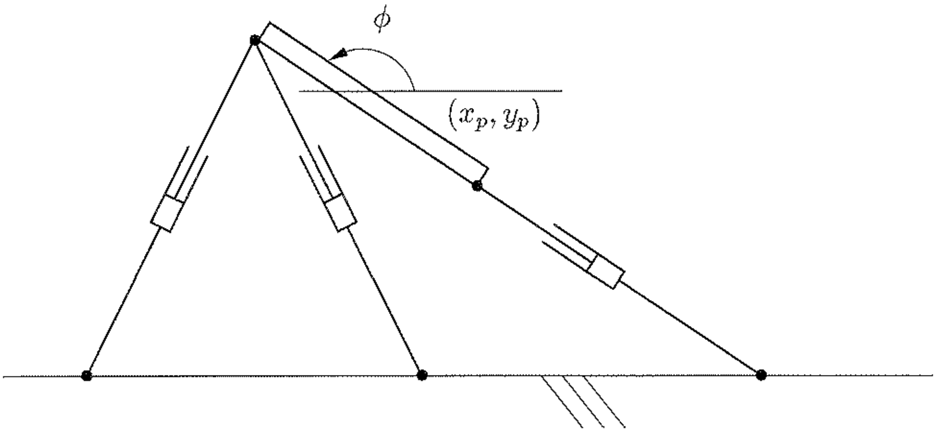


Figure 5.8: Planar parallel platform in a singular position.

Figure 5.9 depicts the computed response of the platform stiffness as the platform rotates through the singular configuration. A definite decrease in the stiffness is encountered near the geometrical singular position. This minimum stiffness is encountered at approximately $\phi = -42°$, and the magnitude of the stiffness is not exactly zero as may be expected.

The reason that the minimum stiffness does not occur at $\phi = -45°$ is that, although the lines along which the actuators work meet at a single point, corresponding to a singular posture at $\phi = -45°$, the stiffening plates are not connected along these same lines. Figure 5.10 depicts the Finite Element model in this kinematically singular posture. It is clear that the stiffening plates are adding to the assembly stiffness in this posture, preventing the stiffness from approaching zero exactly.

5.7 Toolpath planning for an improved planar platform design

Clinton *et al.* [10] suggest an application of stiffness mapping lies in toolpath planning. The placements of identical toolpaths in several areas of the workspace can be evaluated by comparing the maximum machine stiffness and variation in stiffness over the toolpath for each placement.

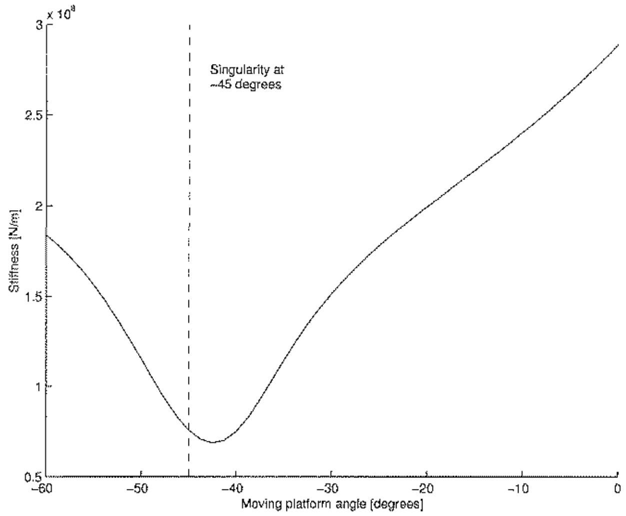
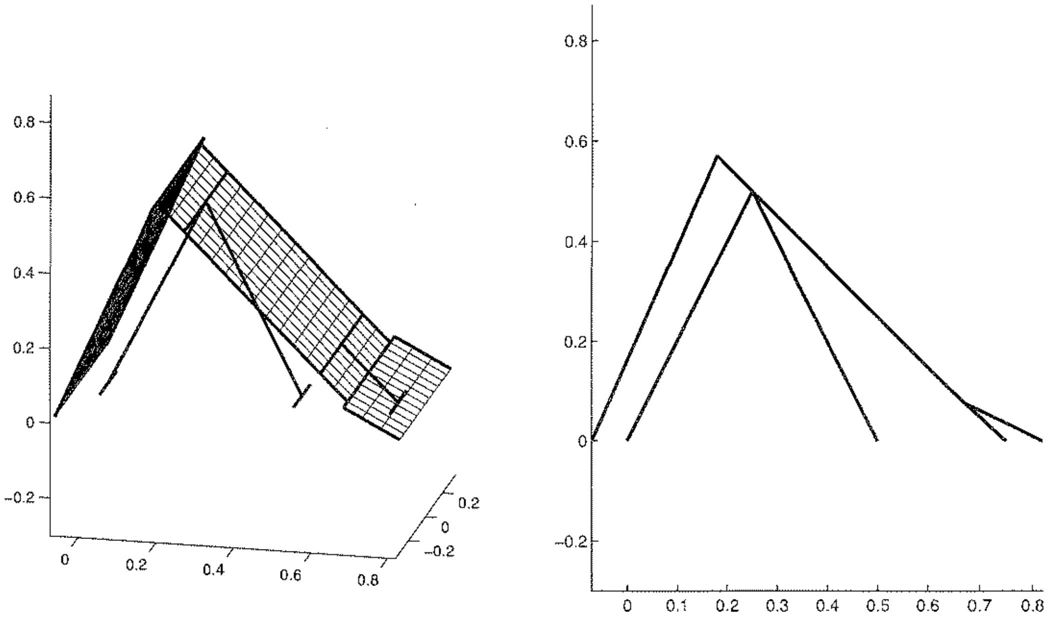


Figure 5.9: Effect of singularity on the real stiffness.

Even thought stiffness is an important factor in toolpath planning, there are other important factors. Examples of criteria used in toolpath planning are: the condition number [9], the so-called quality index [62] and the forces exerted by the actuators (as used by Snyman *et al.* [6]) in executing the prescribed path.



(a) Isometric view of platform in singular configuration.

(b) Side view of platform in singular configuration.

Figure 5.10: Improved platform design in a singular configuration.

Chapter 6

Conclusions and Recommendations

6.1 General remarks

The main objective of the study has been achieved. From the results presented, it is evident that the finally obtained design of the planar parallel platform is indeed feasible for use in machining operations. Using the same tool forces prescribed by Smit [51], an end-effector displacement of 10.2 μm is achieved, with a platform assembly weight of only 285.6 kg. This corresponds to an end-effector stiffness of approximately 2.44×10^8 N/m.

An existing concept design was analyzed and found to be infeasible for accurate machining operations. The major reason for the failure of the original design is the lack of interaction between adjacent stiffening guides. This deficiency has been eliminated in the finally proposed platform design. The stiffness that is achieved by the improved planar parallel platform design compares favorably with the stiffness of the design studied by Clinton *et al.* [10]. This is particularly encouraging since the manipulator studied by Clinton *et al.* (the Ingersoll Octahederal-Hexapod at NIST), is a successful working prototype of a Stewart-platform-based milling machine.

The calculation of out of plane stiffness of a planar parallel manipulator cannot be carried out using traditional robotic stiffness calculations, requiring the determination of the kinematic Jacobian matrix. To date, little or no attention has been paid to the out of plane stiffness of a planar parallel platforms. The calculation of this stiffness requires a finite element analysis (FEA), since the kinematic equation are valid only in the plane of motion. In general it may be noted that FEA has not been used to its full potential in the study of these manipulators. FEA's used by other authors in the field of robotics research are mainly aimed at verifying kinematic robotic stiffness models. These analyses mainly consist of simple truss elements and are, in general, not more accurate than the kinematic models.

In this work the Finite Element Method (FEM) was used to calculate the displacement of the working point. The FEM model makes use of state-of-the-art elements, such as the QCnD-SA flat shell element with drilling degrees of freedom. The use of these elements allows for accurate end-effector displacement calculation and beam-shell interactions.

The study emphasizes the usefulness of mathematical optimization in the design of manipulators. Using mathematical optimization, components which significantly influence the stiffness of the structure, are easily identified. This allows for special attention to be paid to these components in improving the design.

Stiffness maps have been drawn up for the finally obtained design using FEA's. The stiffness maps reveal regions of low stiffness, suggesting the use of stiffness as an aid in toolpath planning. However, a singularity analysis carried out on this design, reveals that the more realistic finite element model adds to the idealized kinematic stiffness of the manipulator. This makes kinematic singularities difficult to predict using more detailed FEA's.

6.2 Performance of optimization algorithms

The study shows that the gradient-based methods are much better suited to this problem than the genetic algorithms (GA's). The usefulness of GA's in problems which are multimodal, discontinuous or nondifferentiable is well known. However, in a problem such as this, where the objective function is relatively smooth and differentiable, not making use of gradient information is a major drawback.

It is shown that the binary GA's have difficulty with unbounded problems or problems with large variable domains. Either a larger number of possible solutions need to be specified or a very coarse sample taken, both of which have obvious disadvantages.

Both a micro GA and a full GA are used to compare performance. Again it is very difficult to suggest which of the two will perform best for a specific problem. This is in agreement with the "No Free Lunch" theorem mentioned at the end of Chapter 4.

The two gradient-based algorithms (LFOPC and Dynamic-Q) proved to be very efficient and robust. This is borne out by the fact that, in the optimization of the proposed improved platform design, even when a badly scaled starting point is chosen, little adverse effect on the performance of the algorithms was experienced.

6.3 Recommendations for future work

This preliminary study into the feasibility of using a planar parallel platform for machining operations proved successful. However, further work is needed before practical machining can be carried out with confidence.

The *validity* of the Finite Element model needs to be further investigated. For example, Clinton *et al.* [10] compared and actively updated their FEM results with displacement measurements taken on a working prototype. Their test procedure is outlined in the ASME standard B5.54 [63].

The optimization should perhaps be extended to take into account *additional constraints* in order to determine beyond any doubt whether or not the design is structurally capable of succeeding. The forces and moments that result in the locking of components such as the actuators and guide rails, are given in catalogs. Since the reaction forces and moments at connection points are calculated during the finite element analysis they may be used in conjunction with the relevant catalog values to determine whether or not locking will occur.

Furthermore, instead of calculating the end-effector displacement at only one point in the workspace, the minimum stiffness over a specified toolpath (or even over the entire workspace) can be used as a criterion for evaluating design performance. This is relatively simple since the infrastructure, which has already been written for this work, can easily be interfaced with kinematic software.

Although in this study, only round beam elements were used in the analyses, the actual shapes of these components are not necessarily round. Taking the actual shapes of the elements into consideration, such as that of the guide rails (which are available only in a discrete number of sizes), will entail a mixed real-integer optimization analysis to which a GA is better suited.

The tool forces used in this study do not take into account the high frequency impact loads that occur during milling operations. These loads are complicated to calculate and are very operation specific, depending on the material being machined, the cut speed and depth, and numerous other factors. Instead of calculating the end-effector displacement for only one specific tool force vector, a *stiffness* requirement for the end-effector should possibly be specified and used as the measure for whether a design is feasible.

The concept *design* itself of the platform also needs further careful consideration. In particular the attachment of the actuators and the stiffening guides and plates need further planning. Only after the design has been finalized should a detailed FEA be performed on the structure to calculate stiffness and structural integrity.

As far as work in the field of Stewart platforms in general goes, Dasgupta *et al.* [26] suggest a number of open problems which are likely to receive attention in the immediate future. Special attention is needed in dynamics and control, workspace and singularity determination, as well as in the study of possible benefits to be gained from the use of redundant manipulators.

Bibliography

[1] University of North Texas: Robotics Glossary. http://www.unt.edu/robotics/glossary-A-D.htm, Accessed June 2001.

[2] University of Texas at Austin: Robotics Glossary. http://www.robotics.utexas.edu/rrg-/glossary/main.htm, Accessed June 2001.

[3] F. Tahmasebi and L.-W. Tsai. Jacobian and stiffness analysis of a novel class of six-dof parallel minimanipulators. In *Proceedings of the 22nd Biennial Mechanisms Conference, ASME*, volume 47, pages 95–102, New York, USA, 1992.

[4] Y. Takeda, H. Funabashi, G. Shen, K. Ichikawa, and K. Hirose. Stiffness analysis of a spatial six-degree-of-freedom in-parallel actuated mechanism with rolling spherical bearings. In *Proceedings Parallel Kinematic Machines 2000 International Conference*, pages 264–273, 13-15 September 2000.

[5] J.A. Snyman, L.J. Du Plessis, and J. Duffy. An optimization approach to the determination of boundaries of manipulator workspaces. *Transactions of ASME, Journal of Mechanical Design*, 122:447–456, 2000.

[6] J.A. Snyman and W.J. Smit. The optimal design of a planar parallel platform for prescribed machining tasks. *Submitted to Multibody System Dynamics*, 2001.

[7] A.M. Hay and J.A. Snyman. The determination of nonconvex workspaces of generally constrained planar Stewart platforms. *Computers and Mathematics with Applications*, 40:1043–1060, 2000.

[8] J.-P. Merlet. Workspace-oriented methodology for designing a parallel manipulator. In *Proccedings IEEE International Conference on Robotics and Automation*, pages 3726–3731, Minneapolis, Minnesota, USA, April 1996.

[9] C. Gosselin. Stiffness mapping for parallel manipulators. *IEEE Transactions on Robotics and Automation*, 6:377–382, 1990.

[10] C.M. Clinton, G. Zhang, and A.J. Wavering. Stiffness modeling of a Stewart-platform-based milling machine. *Transactions of NAMRI/SME*, 15:335–340, 1997.

[11] A. Ibrahimbegovic, R.L. Taylor, and E.L. Wilson. A robust quadrilateral membrane finite element with drilling degrees of freedom. *International Journal for Numerical Methods in Engineering*, 30:445–457, 1990.

[12] A.A. Groenwold and N. Stander. An efficient 4-node 24 d.o.f. thick shell finite element with 5-point quadrature. *Engineering Computations*, 12:723–748, 1995.

[13] A.A. Groenwold and N. Stander. A 24 d.o.f. 4-node flat shell finite element for general unsymmetric orthotropic layered composites. *Engineering Computations*, 15:518–543, 1998.

[14] D.L. Carrol. Chemical laser modeling with genetic algorithms. *AIAA Journal*, 34:338–346, 1996.

[15] J.A. Snyman. The LFOPC leap-frog algorithm for constrained optimization. *Computers and Mathematics with Applications*, 40:1085–1096, 2000.

[16] J.A. Snyman, N. Stander, and W.J. Roux. A dynamic penalty function method for the solution of structural optimization problems. *Applied Mathematical Modelling*, 18:453–460, 1994.

[17] D. Stewart. A platform with six degrees of freedom. *Proceedings of the Institution of Mechanical Engineers 1965-1966*, pages 371–386, 1965.

[18] V.E. Gough. Contribution to discussion to papers on research in automobile stability and control and in tyre performance, by Cornell staff. *Proc. Auto. Div. Instn. Mech. Engrs. 1956-57*, page 392, 1956-57.

[19] V.E. Gough and S.G. Whitehall. Universal tyre test machine. In *Proceedings, Ninth International Technical Congress F.I.S.I.T.A.*, page 117, May 1962.

[20] L.J. Du Plessis. An optimization approach to the determination of manipulator workspaces. Master's thesis, University of Pretoria, Pretoria, South Africa, 1999.

[21] J.-P. Merlet. Parallel manipulators: state of the art and perspectives. *Advanced Robotics*, 8:589–596, 1994.

[22] J.-P. Merlet. Parallel manipulators: state of the art and perspectives. http://www-sop.inria.fr/coprin/equipe/merlet/Etat/etat_de_lart99.html, Accessed February 2001.

[23] Manufacturing Engineering Laboratory (MEL) Gallery Hexapod. http://www.mel.-nist.gov/photos/hexa.html, Accessed May 2001.

[24] PKM (Parallel Kinematic Machines) Bookmarks. http://www.ifw.uni-hannover.de-/robotool/pages/bookmarks.htm, Accessed April 2001.

[25] Z. Ji. Analysis of design parameters in platform manipulators. *Transactions of ASME, Journal of Mechanical Design*, 118:526–531, 1996.

[26] B. Dasgupta and T.S. Mruthyunjaya. The Stewart platform manipulator: a review. *Mechanism and Machine Theory*, 35:15–40, 2000.

[27] B.S. El-Khasawneh and P.M. Ferreira. On using parallel link manipulators as machine tools. *Transactions of NAMRI/SME*, 15:305–310, 1997.

[28] J.-P. Merlet. An algorithm for the forward kinematics of general 6 d.o.f. parallel manipulators. *Technical Report 1331, INRIA, France*, 1990.

[29] M. Raghavan. The Stewart platform of general geometry has 40 configurations. *Transactions of ASME, Journal of Mechanical Design*, 115:277–282, 1993.

[30] C. Innocenti. Direct kinematics in analytical form of the 6-4 fully-parallel mechanism. *Transactions of ASME, Journal of Mechanical Design*, 117:89–95, 1995.

[31] W. Lin, M. Griffis, and J. Duffy. Forward displacement analysis of the 4-4 Stewart platforms. *Transactions of the ASME, Journal of Mechanical Design*, 114:444–450, 1992.

[32] J.-P. Merlet. Direct kinemetics and assembly modes of parallel manipulators. *The International Journal of Robotics Research*, 11:150–162, 1992.

[33] C. Gosselin, J. Sefrioui, and M. Richard. Solutions polynomiales au probleme de la cinematique directe des manipulateurs paralleles plans a trois degres de liberte. *Mechanism and Machine Theory*, 27:107–119, 1992.

[34] E.F. Fichter. A Stewart platform-based manipulator: general theory and practical construction. *The International Journal of Robotics Research*, 5:157–182, 1986.

[35] B. Dasgupta and T.S. Mruthyunjaya. Singularity-free path planning for the Stewart platform manipulator. *Mechanism and Machine Theory*, 33:711–725, 1998.

[36] B. Dasgupta, S. Reddy, and T.S. Mruthyunjaya. Synthesis of a force-torque sensor based on the Stewart platform mechanism. In *Proceedings of National Convension on Industrial Problems in Machines and Mechanisms (IPROMM)*, pages 14–23, Bangalore, India, 1995.

[37] J.-P. Merlet. Direct kinematics of planar parallel manipulators. In *Proceedings IEEE International Conference on Robotics and Automation*, pages 3744–3749, Minneapolis, Minnesota, USA, April 1996.

[38] S.M. Satya, P.M. Ferreira, and M.W. Spong. Hybrid control of a planar 3-DOF parallel manipulator for machining operations. *Transactions of NAMRI/SME*, 23:273–280, 1995.

[39] C.A. Klein and B.E. Blaho. Dexterity measures for the design and control of kinematically redundant manipulators. *IEEE Journal of Robotics and Automation*, 6:72–83, 1987.

[40] B.S. El-Khasawneh and P.M. Ferreira. Computation of stiffness and stiffness bounds for parallel link manipulators. *Submitted to the International Journal for Machine Tools and Manufacturing.*

[41] X.-J. Liu, Z.-L. Jin, and F. Gao. Otimum design of 3-DOF sperical parallel manipulators with respect to the conditioning and stiffness indices. *Mechanism and Machine Theory*, 35:1257–1267, 2000.

[42] F. Tahmasebi and L.-W. Tsai. On the stiffness of a novel six-dof parallel mini-manipulator. *Journal of Robotic Systems*, 12(12):845–856, 1995.

[43] L.J. Du Plessis, J.A. Snyman, and W.J. Smit. A configurationally adjustable planar Stewart platform machining center with feasible and optimum placement of workspace relative to toolpath. In *ASME 2000 Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Baltimore, Maryland, USA, 10-13 September 2000.

[44] R.S. Ball. *A Treaties on the Theory of Screws.* Cambridge: Cambridge University Press, 1900.

[45] K.J. Waldron and K.H. Hunt. Series-parallel dnalities in actively coodinated mechanisms. *The International Journal of Robotics Research*, 10:473–480, 1991.

[46] E.L. Wilson. EDSAP - an EDucational version of a three degree dimensional Structural Analysis Program. Computer code, 1988.

[47] T.R. Chandrupatla and A.D. Belegundu. *Introduction to Finite Elements in Engineering.* Prentice Hall, second edition, 1997.

[48] K.-J. Bathe. *Finite element procedures.* Prentice-Hall, Inc., 1996.

[49] O.C. Zienkiewicz and R.L. Taylor. *The Finite Element Method: Basic formulation and linear problems*, volume II. McGraw-Hill, London, 1991.

[50] K.-J. Bathe and E.N. Dvorkin. A four node plate bending element based on Mindlin-Reissner plate theory and a mixed interpolation. *International Journal for Numerical Methods in Engineering*, 21:367–383, 1985.

[51] Smit W.J. The optimal design of a planar Stewart platform for prescribed machining tasks. Master's thesis, University of Pretoria, 1999.

[52] E.J. Haug, C.-M. Luh, F.A. Adkins, and J. Wang. Numerical algorithms for mapping boundaries of manipulator workspaces. *Transactions of ASME, Journal of Mechanical Design*, 118:228–234, 1996.

[53] D. Wolpert and W. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1:67–82, 1997.

[54] T.M. English. Evaluation of evolutionary and genetic operators: No free lunch. In L.J. Fogel, P.J. Angeline, and T. Baeck, editors, *Evolutionary Programming V: Proc. Fifth Annual Conference on Evolutionary Programming*, pages 163–169, San Diego, CA, USA, February 1995.

[55] J.A. Snyman. A new dynamic method for unconstrained minimization. *Applied Mathematical Modelling*, 6:449–462, 1982.

[56] J.A. Snyman. An improved version of the original leap-frog dynamic method for unconstrained minimization LFOP1(b). *Applied Mathematical Modelling*, 7:216–218, 1983.

[57] J.A. Snyman and A.M. Hay. The Dynamic-Q optimization method: an alternative to SQP? In *Proceedings of the Inernational Workshop on Multidisciplinary Design Optimization. (Also Submitted to Computers and Mathematics with Applications 2001)*, pages 163–172, Pretoria, South Africa, 7-10 August 2000.

[58] K.J. Craig, D.J. De Kock, and J.A. Snyman. Using CFD and mathematical optimization to minimize stack pollution. *International Journal for Numerical Methods in Engineering*, 44:551–566, 1999.

[59] K.J. Craig, P. Venter, D.J. De Kock, and J.A. Snyman. Optimization of structured grid spacing parameters for seperated flow simulation using mathematical optimization. *Journal of Wind Engineering and Industrial Aerodynamics*, 80:221–231, 1999.

[60] D.J. De Kock, K.J. Craig, and J.A. Snyman. Using mathematical optimization in CFD analysis of a continuous quenching process. *International Journal for Numerical Methods in Engineering*, 47:985–999, 2000.

[61] L.J. Du Plessis. Personal communication. University of Pretoria, Pretoria, South Africa, 2001.

[62] J. Lee, J. Duffy and M. Keler. The optimum quality index for the stability of in-parallel planar platform devices. In *Proceedings of the 1996 ASME Design Engineering Technical Conferences and Computers in Engineering Conference*, pages 1–7, 1996.

[63] ANSI/ASME B5.54-1992. Methods for performance evaluation of computer controlled machining centers. *ASME, New York*, 1993.

[64] J. Duffy. *Statics and kinematics with applications to robotics.* Cambridge University Press, 1996.

[65] D. Goldberg. *Genetic algorithms in search, optimization, and machine learning.* Addison-Wesley, 1989.

[66] Z. Michaelwicz. *Genetic algorithms + data structure = evolution programs.* Springer-Verlag, 1992.

[67] C.R. Houck, J.A. Joines, and M.G. Kay. A genetic algorithm for function optimization: A Matlab implementation, NCSU-IE TR 95-09. http://www.ie.ncsu.edu/mirage-/GAToolBox/gaot/, Accessed April 2001.

[68] J. Holland. *Adaption in natural and artificial systems.* The University of Michigan Press, Ann Arbor, USA, 1975.

[69] T. Bäck. Selective pressure in evolutionary algorithms: A characterization of selection mechanisms. In *Proc. 1^{st} IEEE Conf. on Evolutionary Computation*, pages 57–62, Pinscataway, USA, 1994.

[70] D.L. Carrol. GAFortran Version 1.7. Computer code, 1998.

[71] H. Dubbel. *Handbook of mechanical engineering.* Springer-Verlag, London, 1994.

Appendix A

Derivation of robotic stiffness using the kinematic Jacobian

In this appendix the in-plane stiffness matrix of a planar parallel platform will be derived using the kinematic Jacobian. This derivation stems mostly from the book of Duffy [64], and the notation here is consistent with his.

A.1 Introductory concepts

Consider a line drawn in the XY plane as depicted in Figure A.1.

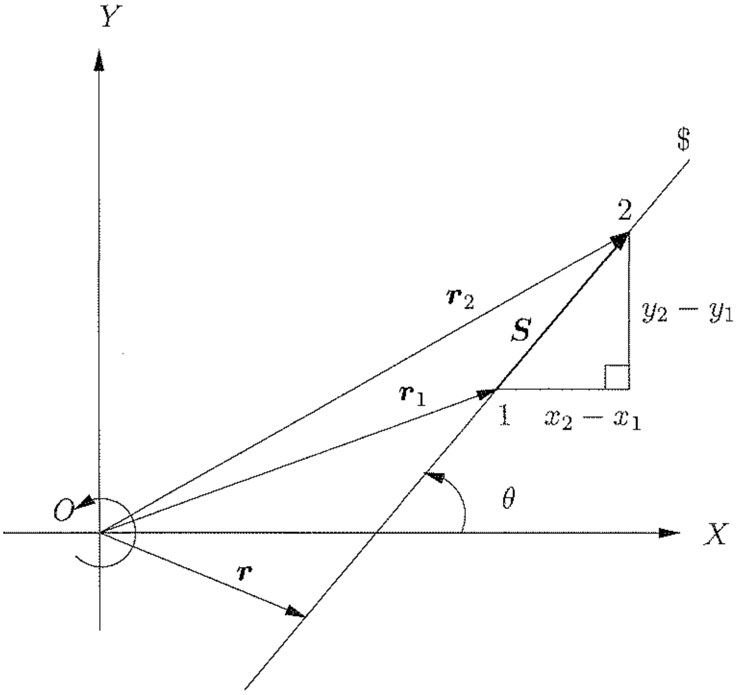


Figure A.1: A line in the XY plane.

The two vectors $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ from O to points 1 and 2, with coordinates (x_1, y_1) and (x_2, y_2) determine the directed line segment 12 with vector $\boldsymbol{S}$ given by

$$\boldsymbol{S} = \boldsymbol{r}_2 - \boldsymbol{r}_1 \tag{A.1}$$

The projections of $\boldsymbol{S}$ onto the X and Y axes are, respectively,

$$L = x_2 - x_1 = |\boldsymbol{S}|\cos\theta \quad \text{and} \quad M = y_2 - y_1 = |\boldsymbol{S}|\sin\theta \tag{A.2}$$

The moment of the directed line segment 12 about the origin is given by the vector cross product $\boldsymbol{r} \times \boldsymbol{S}$, where $\boldsymbol{r}$ is any vector drawn from O to any point on the straight line joining points 1 and 2, and the vector $\boldsymbol{S}$ can be located anywhere on the line.

It can be shown that the equation of the straight line can be written as

$$Ly - Mx + R = 0 \tag{A.3}$$

where

$$R = \begin{vmatrix} x_1 & y_1 \\ x_2 & y_2 \end{vmatrix} \tag{A.4}$$

and that

$$\boldsymbol{r} \times \boldsymbol{S} = R\boldsymbol{k} \tag{A.5}$$

The three numbers L, M and R were first established by Plücker, and are known as the Plücker line coordinates. They are homogeneous because substituting λL, λM and λR, where λ is a nonzero scalar into (A.3) yields the same line. However, their units are not consistent. L and M have dimensions of $(\text{length})^1$, while R has dimensions of $(\text{length})^2$. Because of this lack of consistency in dimensions, the coordinates are represented by the ordered triple of real numbers $\{L, M; R\}$. A line segment which is determined by the ordered triple of real numbers $\{L, M; R\}$ is known as a *line bound vector*.

For a unit line segment where $|\boldsymbol{S}| = \sqrt{L^2 + M^2} = 1$, it can be shown that $R = p$ and the line coordinates are $\{c, s; p\}$, with abbreviations $c = \cos\theta$ and $s = \sin\theta$ introduced, and where

$$p = \frac{R}{\sqrt{L^2 + M^2}}. \tag{A.6}$$

p is the length of the vector drawn from O perpendicular to the line \$.

From this point onward, let $\boldsymbol{S}$ represent a unit vector in along \$ with $|\boldsymbol{S}| = 1$, and let us define $\boldsymbol{S}_O = \boldsymbol{r} \times \boldsymbol{S}$.

The coordinates $\hat{s}$ for a line \$ can be expressed by the ordered pair of vectors $\{\boldsymbol{S}; \boldsymbol{S}_0\}$ where

$$\hat{s} = \{\boldsymbol{S}; \boldsymbol{S}_0\} \tag{A.7}$$

In this representation, the coordinates $\hat{s}$ are defined as *ray coordinates* because the line drawn through two points may be imagined to be an actual narrow ray of light originating at a point.

The coordinates for the same line \$ can be expressed equally by the ordered pair of vectors $\{\boldsymbol{S}_0; \boldsymbol{S}\}$ and denoted by $\hat{S}$, where

$$\hat{S} = \{\boldsymbol{S}_0; \boldsymbol{S}\} \tag{A.8}$$

The coordinates of $\hat{S}$ are defined as *axis coordinates*, because the line \$ is considered as the meet of two planes. Therefore the lower case $\hat{s}$ and the upper case $\hat{S}$ distinguish the two coordinate representations of the same line \$.

A.2 Mutual moment

Figure A.2 illustrates a line $\$_i$ in the XY plane with ray coordinates

$$\hat{s} = \begin{bmatrix} c_i \\ s_i \\ r_i + r_i' \end{bmatrix} \tag{A.9}$$

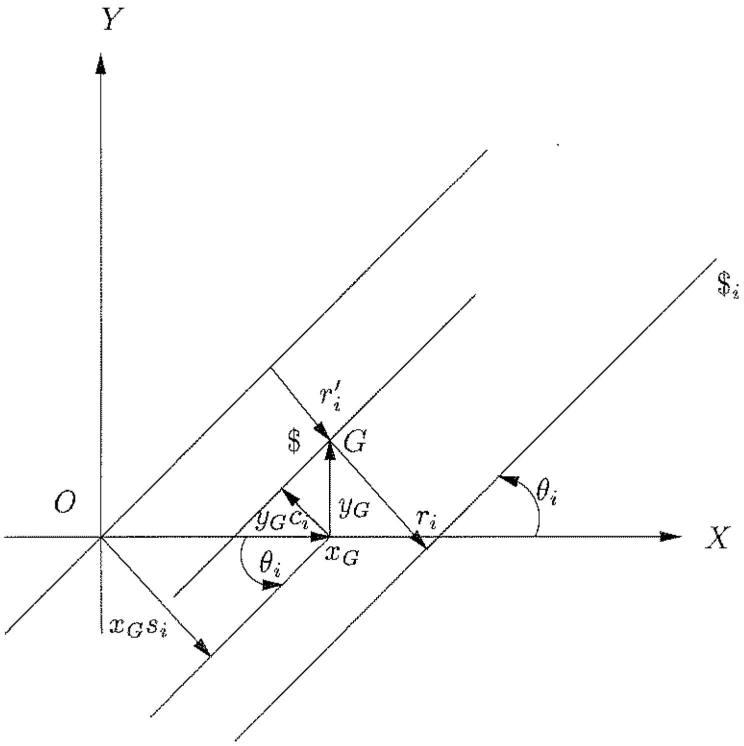


Figure A.2: Mutual moment of two lines.

Assume that there is a second line \$, drawn through the point G and perpendicular to the XY plane with axis coordinates

$$\hat{S} = \begin{bmatrix} y_G \\ -x_G \\ 1 \end{bmatrix} \tag{A.10}$$

The *mutual moment* is defined as $\hat{s}_i^T \hat{S} (= \hat{S}^T \hat{s}_i)$. From (A.9) and (A.10),

$$\begin{aligned} \hat{s}_i^T \hat{S} &= [c_i, s_i; r_i + r_i'] \begin{bmatrix} y_G \\ -x_G \\ 1 \end{bmatrix} \\ &= r_i + r_i' - (x_G s_i - y_G c_i). \end{aligned} \tag{A.11}$$

From Figure A.2, $(x_G s_i - y_G c_i) = r_i'$ and therefore,

$$\hat{s}_i^T \hat{S} = r_i \tag{A.12}$$

The mutual moment of this pair of normalized lines, which are mutually perpendicular, is their common perpendicular distance r_i.

A.3 Infinitesimal displacement analysis of a planar parallel manipulator

Figure A.3 illustrates the moving platform of a parallel manipulator undergoing an infinitesimal rotation $\delta\phi$ about an axis \$. This is modeled by a revolute joint at G to which the moving platform is rigidly connected. The lines $\$_{iB}$ and $\$_{iC}$ are perpendicular to the lines of the connector $\$_i$, and they pass through the fixed and moving pivots B_i and C_i, respectively.

When the moving platform rotates relative to the fixed platform about an axis through G, each moving pivot C_i displaces to point C_i'. This displacement can be decomposed into two displacements, an infinitesimal displacement δl_i along the line $\$_i$ together with an infinitesimal displacement $l_i \delta\theta_i$ which is tangent to a circle of radius l_i centered at B_i as shown in Figure A.4. It can be shown from this figure that

$$\delta l_i = r_i \delta\phi \quad \text{and} \quad l_i \delta\theta_i = r_{iC} \delta\phi \tag{A.13}$$

where r_i and r_{iC} are the perpendicular distances from G to the lines $\$_i$ and $\$_{iC}$ respectively. Therefore, as shown in (A.12)

$$r_i = \hat{s}_i^T \hat{S}, \quad \text{and} \quad r_{iC} = \hat{s}_{iC}^T \hat{S} \tag{A.14}$$

where $\hat{s}_i$, $\hat{s}_{iC}$ and $\hat{S}$ are the coordinates of the lines $\$_i$, $\$_{iC}$ and the axis \$ through G. If (A.14) is substituted into (A.13), we get

$$\delta l_i = \hat{s}_i^T (\delta\phi \hat{S}) = \hat{s}_i^T \delta\hat{D}, \; i = 1,2,3 \quad \text{and} \quad l_i \delta\theta_i = \hat{s}_{iC}^T (\delta\phi \hat{S}) = \hat{s}_{iC}^T \delta\hat{D}, \; i = 1,2,3 \tag{A.15}$$

(A.15) can now be expressed in matrix from as:

$$\begin{bmatrix} \delta l_1 \\ \delta l_2 \\ \delta l_3 \end{bmatrix} = j^T \delta \hat{D} \text{ and } \begin{bmatrix} l_1\delta\theta_1 \\ l_2\delta\theta_2 \\ l_3\delta\theta_3 \end{bmatrix} = [C]^T \delta \hat{D} \tag{A.16}$$

If the reference point O is chosen on the lower fixed platform, the coordinates of the lines $\$_i$, $\$_{iB}$ and $\$_{iC}$ can easily be determined, and the matrices for the lines $\$_{iB}$ and $\$_{iC}$ (perpendicular to $\$_i$) and passing through pivots B and C respectively) are:

$$[B] = \begin{bmatrix} -s_1 & -s_2 & -s_3 \\ c_1 & c_2 & c_3 \\ q_{1B} & q_{2B} & q_{3B} \end{bmatrix} \quad \text{and} \quad [C] = \begin{bmatrix} -s_1 & -s_2 & -s_3 \\ c_1 & c_2 & c_3 \\ q_{1C} & q_{2C} & q_{3C} \end{bmatrix} \tag{A.17}$$

where q_{iB} and q_{iC} are the perpendicular distances from O to the lines $\$_{iB}$ and $\$_{iC}$ respectively, and $q_{iC} = q_{iB} + l_i$.

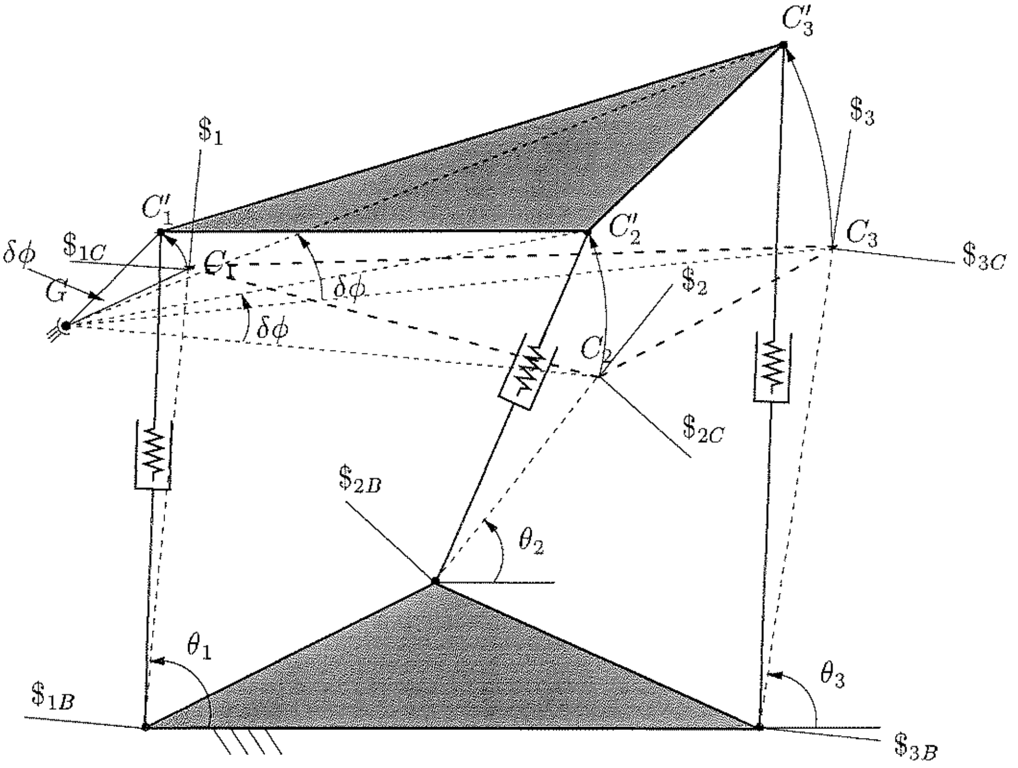


Figure A.3: Infinitesimal rotation of the moving platform about point G

Observe from Figures A.3 and A.4 that when the moving platform rotates through an angle $\delta\phi$, each connector moves from a line $\$_i$ to a new line $\$_i^+$ due to a small rotation $\delta\theta_i$ about the fixed pivot B_i. We can now determine the coordinates of the new line $\$_i'$.

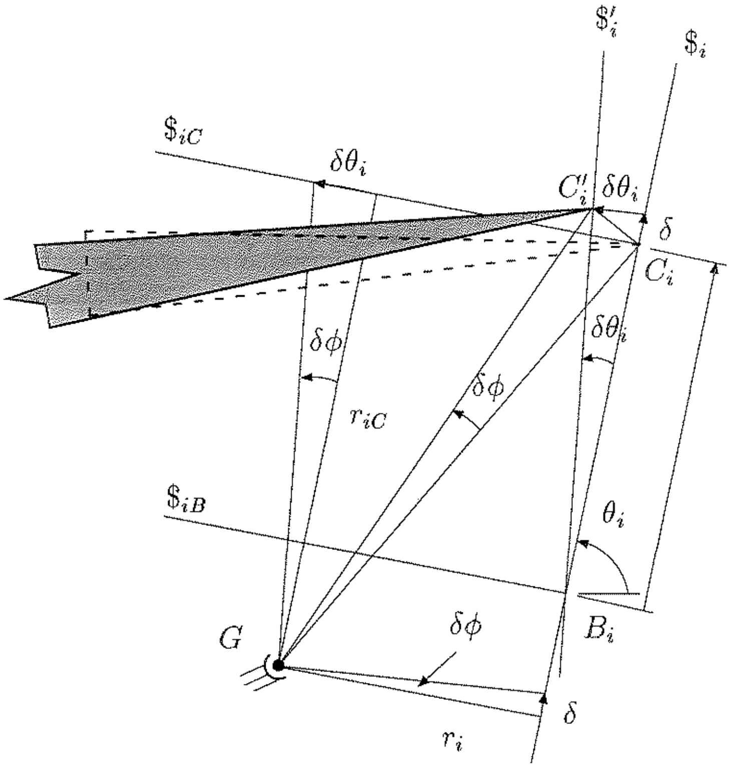


Figure A.4: Infinitesimal rotation of a single connector

A.4 The differential of a line

Figure A.5 represents a single connector which is attached to a revolute joint at B_i, and the revolute joint is in turn fixed to ground. If the connector rotates a small amount $\delta\theta_i$, the

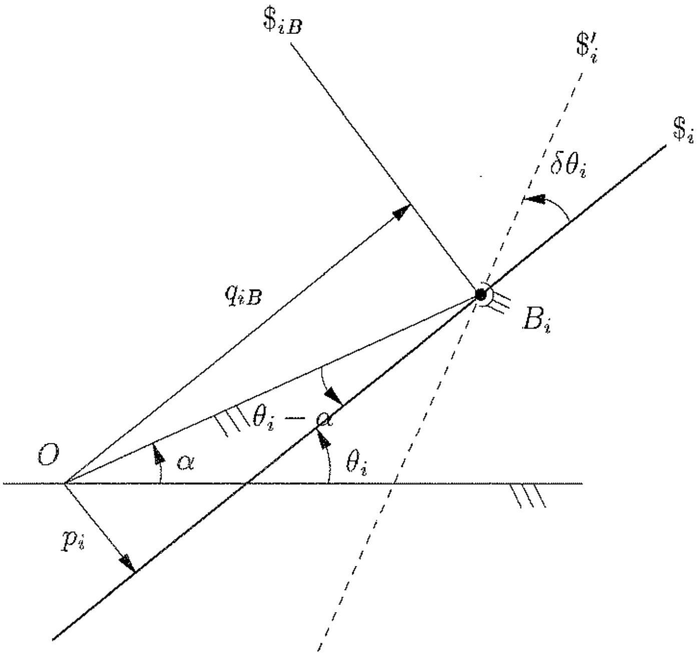


Figure A.5: Differential of a single connector

line $\$_i$ rotates to a new location $\$_i'$.

Since $\hat{s}_i$, the coordinates of the line $\$_i$ are a function of the angle only,

$$\hat{s}_i = \hat{s}_i(\theta_i), \tag{A.18}$$

the differential of $\hat{s}_i$ is given by

$$\delta \hat{s}_i = \frac{d\hat{s}_i}{d\theta_i}\delta\theta_i \tag{A.19}$$

The coordinates of the line $\$_i$ are given by:

$$\hat{s}_i = \{c_i, s_i; p_i\} = \{c_i, s_i; OB\text{sin}(\theta_i - \alpha)\} \tag{A.20}$$

where $c_i = \cos\theta_i$ and $s_i = \sin\theta_i$. Now we define

$$\hat{s}_{iB} = \frac{d\hat{s}_i}{d\theta_i} = \{-s_i, c_i; OB\text{cos}(\theta_i - \alpha)\} = \{-s_i, c_i; q_{iB}\} \tag{A.21}$$

These are the coordinates for line $\$_{iB}$, which is defined as the *geometrical differential* of $\$_i$. The coordinates of the line $\$_i'$ can now be written as

$$\hat{s}_i' = \hat{s}_i + \delta\hat{s}_i = \hat{s}_i + \hat{s}_{iB}\delta\theta_i, \tag{A.22}$$

which are a linear combination of the coordinates of lines $\$_i$ and $\$_{iB}$.

A.5 Stiffness of a planar parallel platform

Figure A.6 represents a relatively general planar parallel platform with compliance introduced in the connectors. The lower platform is fixed to ground, while the upper platform is manipulated by the compliant actuators. The actuators are represented by revolute joint - prismatic connector - revolute joint (RPR) connectors, each containing a linear spring.

The externally applied load $\boldsymbol{f}$ is expressed using the coordinates $\{L, M; R\}$ with reference to some coordinate system. The value for R is dependent on the placement of the origin O.

The force can be expressed in vector by the coordinates

$$\hat{w} = \{\boldsymbol{f}, \boldsymbol{c}_O\} \tag{A.23}$$

The two representations are related as follows:

$$\boldsymbol{f} = (L\boldsymbol{i} + M\boldsymbol{j}), \quad \boldsymbol{c}_0 = R\boldsymbol{k} \tag{A.24}$$

$\boldsymbol{f}$ can also be represented as a multiple of a unit vector $\boldsymbol{S}$ in the form $f\boldsymbol{S}$ where $|\boldsymbol{S}| = 1$ and $f = |\boldsymbol{f}|$, and the moment vector can thus be represented as $\boldsymbol{c}_O = f\boldsymbol{S}_O$ where $\boldsymbol{S}_O = \boldsymbol{r} \times \boldsymbol{S}$.

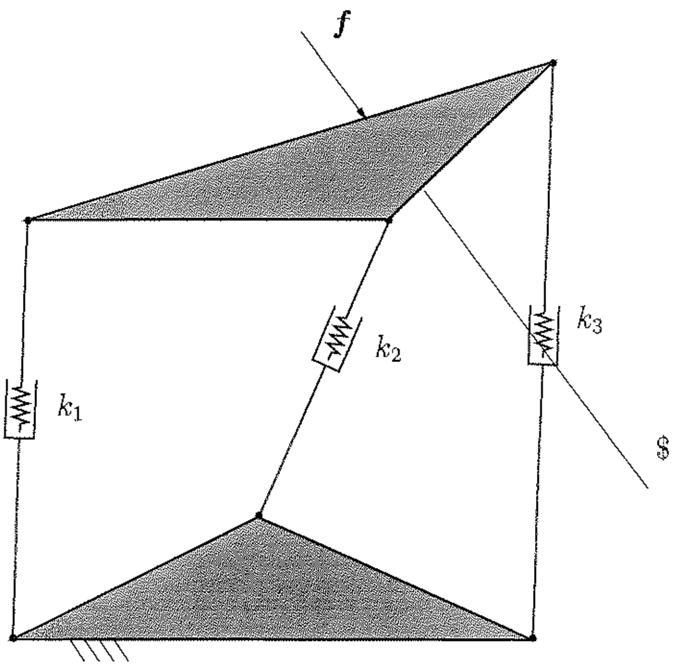


Figure A.6: Compliant planar parallel platform.

Since $\boldsymbol{S}$ is a unit vector, we may use the notation

$$\hat{s} = \{\boldsymbol{S}; \boldsymbol{S}_O\} \tag{A.25}$$

In this way, the coordinates for a force can be expressed by a scalar f (the magnitude of the force) and a unit line vector as:

$$\hat{w} = f\hat{s} \tag{A.26}$$

If the moving platform is at rest, the externally applied load must be in equilibrium with the resultant forces in the actuators. This equilibrium condition can be mathematically represented in the form

$$\hat{w} = f_1\hat{s}_1 + f_2\hat{s}_2 + f_3\hat{s}_3 \tag{A.27}$$

where the externally applied force has coordinates $\hat{w}$ and magnitude f, and is applied along the line \$. f_1, f_2 and f_1 are the magnitudes of the resultant forces in the actuators and $\hat{s}_1$, $\hat{s}_2$ and $\hat{s}_3$ are the line coordinates for the connectors.

Since the force in each of the actuators is given by:

$$f_i = k_i(l_i - l_{0i}), \qquad i = 1, 2, 3 \tag{A.28}$$

where l_{0i} is the free length of the springs, l_i is the length of the loaded spring and k_i is the stiffness constant of the spring. If the coordinates of $\hat{w}$ are known as well as $\hat{s}_1$, $\hat{s}_2$, $\hat{s}_1$, the coordinates of lines $\$_1$, $\$_2$ and $\$_3$ respectively, (A.27) can be written as

$$\hat{w} = k_1(l_1 - l_{01})\hat{s}_1 + k_2(l_2 - l_{02})\hat{s}_2 + k_3(l_3 - l_{03})\hat{s}_3 \tag{A.29}$$

In order to calculate the stiffness of the assembly, a small change ($\delta\hat{w}$) in the external load is applied, resulting in an infinitesimal rotation $\delta\phi$ about an axis \$ through point G and perpendicular to the page, as shown in Figure A.3.

Using the results from Sections A.3 and A.4, the effect of the small change $\delta\hat{w}$ in the externally applied load can now be calculated.

$$\begin{aligned} \delta\hat{w} &= k_1\delta l_1\hat{s}_1 + k_2\delta l_2\hat{s}_2 + k_3\delta l_3\hat{s}_3 + k_1(l_1 - l_{01})\frac{d\hat{s}_1}{d\theta_1}\delta\theta_1 + \\ & \quad k_2(l_2 - l_{02})\frac{d\hat{s}_2}{d\theta_2}\delta\theta_2 + k_3(l_3 - l_{03})\frac{d\hat{s}_3}{d\theta_3}\delta\theta_3 \end{aligned} \tag{A.30}$$

We now define $\rho_i = \frac{l_{0i}}{l_i}$ and using the result from (A.21) where $\frac{d\hat{s}_i}{d\theta_i} = \hat{s}_{iB}$, (A.30) becomes:

$$\begin{aligned} \delta\hat{w} &= \hat{s}_1k_1\delta l_1 + \hat{s}_2k_2\delta l_2 + \hat{s}_3k_3\delta l_3 + \hat{s}_{1B}k_1(1-\rho)l_1\delta\theta_1 + \\ & \quad \hat{s}_{2B}k_2(1-\rho)l_2\delta\theta_2 + \hat{s}_{3B}k_3(1-\rho)l_3\delta\theta_3 + \end{aligned} \tag{A.31}$$

Finally, (A.31) can be written in matrix from as:

$$\delta\hat{w} = [\hat{s}_1\ \hat{s}_2\ \hat{s}_3][k]\begin{bmatrix} \delta l_1 \\ \delta l_2 \\ \delta l_3 \end{bmatrix} + [\hat{s}_{1B}\ \hat{s}_{2B}\ \hat{s}_{3B}][k(1-\rho)]\begin{bmatrix} l_1\delta\theta_1 \\ l_2\delta\theta_2 \\ l_3\delta\theta_3 \end{bmatrix} \tag{A.32}$$

$$\text{where } [k] = \begin{bmatrix} k_1 & 0 & 0 \\ 0 & k_2 & 0 \\ 0 & 0 & k_3 \end{bmatrix} \text{ and } [k(1-\rho)] = \begin{bmatrix} k_1(1-\rho) & 0 & 0 \\ 0 & k_2(1-\rho) & 0 \\ 0 & 0 & k_3(1-\rho) \end{bmatrix} \tag{A.33}$$

Also, it is known that:

$$j = [\hat{s}_1\ \hat{s}_2\ \hat{s}_3] = \begin{bmatrix} c_1 & c_2 & c_3 \\ s_1 & s_2 & s_3 \\ p_1 & p_2 & p_3 \end{bmatrix} \tag{A.34}$$

is the matrix of the coordinates of the lines $\$_1$, $\$_2$ and $\$_3$, and

$$[B] = [\hat{s}_{1B}\ \hat{s}_{2B}\ \hat{s}_{3B}] = \begin{bmatrix} -s_1 & -s_2 & -s_3 \\ c_1 & c_2 & c_3 \\ q_{1B} & q_{2B} & q_{3B} \end{bmatrix} \tag{A.35}$$

is the matrix of the coordinates of the lines $\$_{1B}$, $\$_{2B}$ and $\$_{3B}$. Therefore, given (A.16), (A.32) simplifies to

$$\delta w = \{j[k]j^T + [B][k(1-\rho)][C]^T\}\delta\hat{D} \tag{A.36}$$

which can be expressed in the usual fashion:

$$\delta\hat{w} = [K]\delta\hat{D} \tag{A.37}$$

where the stiffness matrix $[K]$ is given by

$$[K] = j[k]j^T + [B][k(1-\rho)][C]^T \tag{A.38}$$

$[K]$ is symmetrical only at the unloaded position for which $\rho_i = 1$ or $l_i = l_{0i}$ and $[K] = j[k]j^T$.

Appendix B

Beam and frame elements

In this appendix, the theory and finite element formulation of beam and frame structures are considered. The formulation closely resembles the treatment by Chandrupatla and Belegungu [47], and the notation used is similar.

A 'beam' is a mathematical approximation to a slender member that is used for supporting transverse loading, while complex structures with rigidly connected members (beams) are called 'frames'.

Only beams with symmetric cross sections with respect to the plane of loading ($x - y$ plane in Figure B.2) are considered here. A relatively general horizontal beam is shown in Figure B.1, while Figure B.2 shows the cross section and bending stress distribution. For small deflections, and from elementary beam theory, we recall that

$$\sigma = \frac{-My}{I} \tag{B.1}$$

$$\epsilon = \frac{\sigma}{E} \tag{B.2}$$

$$\frac{d^2v}{dx^2} = \frac{M}{EI} \tag{B.3}$$

where σ is the normal stress, ϵ is the normal strain, M is the bending moment at the section, v is the deflection of the centroid axis at x and I is the moment of inertia of the section about the neutral axis (z axis passing through the centroid).

B.1 Galerkin Approach

For the Galerkin formulation, we start from the equilibrium of an elemental length. From Figure B.3, it is clear that

$$\frac{dV}{dx} = p \tag{B.4}$$

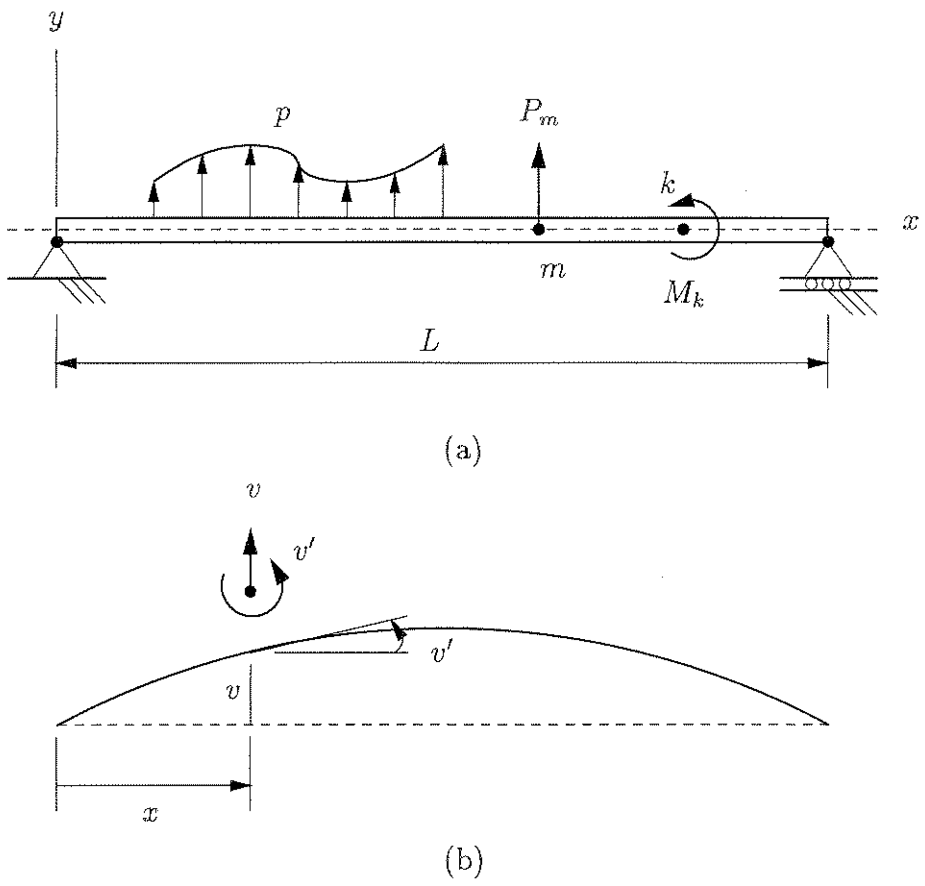


Figure B.1: (a) General beam loading and (b) deformation of neutral axis.

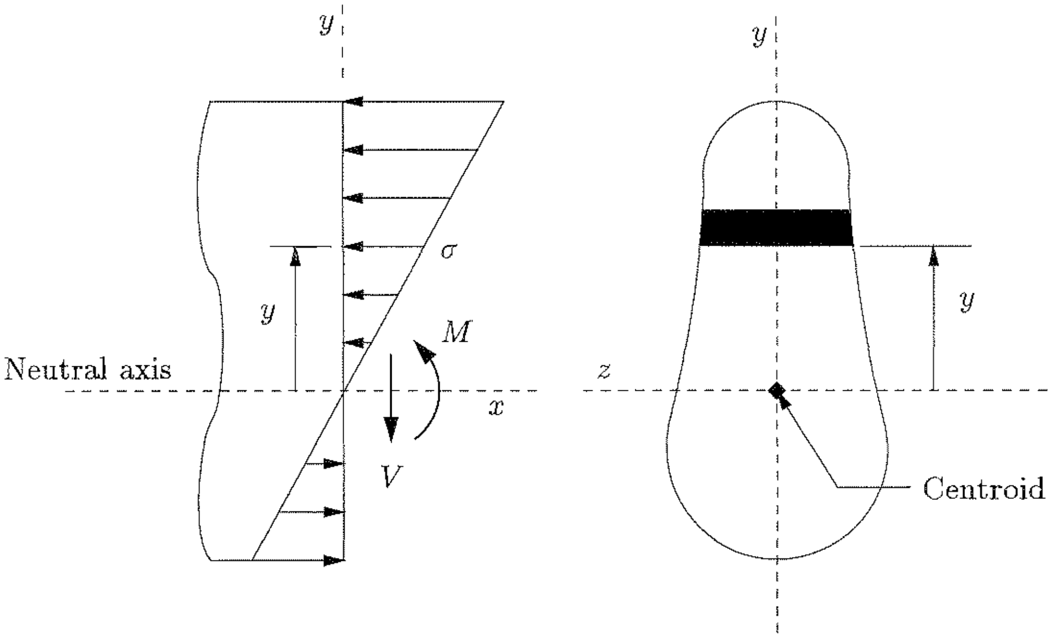


Figure B.2: Beam section and stress distribution.

$$\frac{dM}{dx} = V \tag{B.5}$$

Combining (B.3), (B.4) and (B.5), the equilibrium equation is given by

$$\frac{d^2}{dx^2}\left(EI\frac{d^2v}{dx^2}\right) - p = 0 \tag{B.6}$$

The approximate solution v is required (constructed by finite element shape functions) such that

$$\int_0^L \left[\frac{d}{dx^2}\left(EI\frac{d^2v}{dx^2}\right) - p\right]\phi \, dx = 0 \tag{B.7}$$

where ϕ is an arbitrary function using the same basis function as v. ϕ is required to satisfy the natural boundary conditions, thus ϕ is zero where v is specified as being zero. Now, integrating the first term of (B.7) by parts, and splitting the integral from 0 to L into intervals from 0 to x_m, x_m to x_k and x_k to L, we obtain

$$\begin{aligned}\int_0^L EI\frac{d^2v}{dx^2}\frac{d^2\phi}{dx^2}dx - \int_0^L p\phi \, dx + \frac{d}{dx}\left(EI\frac{d^2v}{dx^2}\right)\phi\,\bigg|_0^{x_m} + \frac{d}{dx}\left(EI\frac{d^2v}{dx^2}\right)\phi\,\bigg|_{x_m}^{L} & \\ - EI\frac{d^2v}{dx^2}\frac{d\phi}{dx}\bigg|_0^{x_k} - EI\frac{d^2v}{dx^2}\frac{d\phi}{dx}\bigg|_{x_k}^{L} &= 0\end{aligned} \tag{B.8}$$

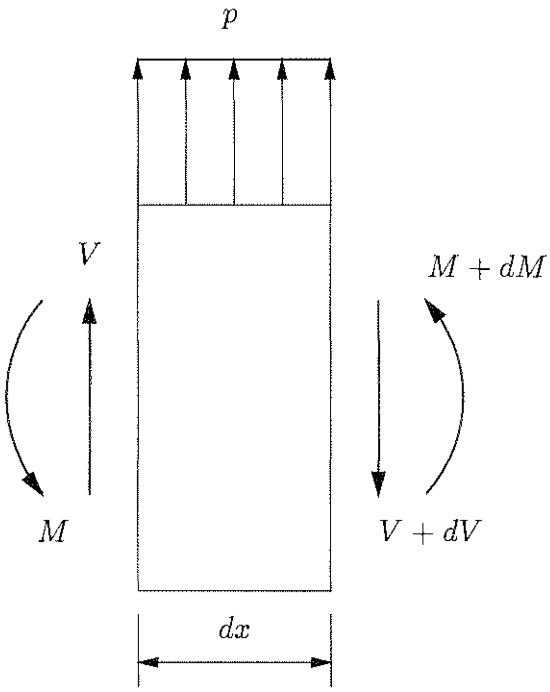


Figure B.3: Free body diagram of an elemental length dx.

We note that $EI(d^2v/dx^2)$ is equal to the bending moment M from (B.3). Also note that $(d/dx)[EI(d^2v/dx^2)]$ is equal to the shear force V from (B.5). From the beam depicted in Figure B.1, ϕ and M are zero at the supports. At x_m, the jump in shear force is P_m and at x_k, the jump in bending moment is $-M_k$. Thus, we get

$$\int_0^L EI\frac{d^2v}{dx^2}\frac{d^2\phi}{dx^2}dx - \int_0^L p\phi\, dx - \sum_m P_m\phi_m - \sum_k M_k\phi'_k = 0 \tag{B.9}$$

For the finite element formulation based on Galerkin's approach, v and ϕ are constructed using the same shape functions. Equation B.9 is precisely the statement of the principal of virtual work.

B.2 Finite element formulation

For a single two noded element, the local degrees of freedom are represented by

$$\boldsymbol{q} = [q_1 \quad q_2 \quad q_3 \quad q_4]^T \tag{B.10}$$

where $\boldsymbol{q}$ is the same as $[v_1 \quad v'_1 \quad v_2 \quad v'_2]^T$. The shape functions for interpolating v are defined in terms of the local coordinate ξ on -1 to 1, as shown in Figure B.4. Since nodal values and nodal slopes are involved, Hermitian shape functions are defined. The shape functions are of cubic order, as represented below

$$H_i = a_i + b_i\xi + c_i\xi^2 + d_i\xi^3 \qquad i = 1,2,3,4 \tag{B.11}$$

(The coefficients a_i, b_i, c_i, d_i, obtained by imposing the required conditions, are shown in Figure B.4.) This yields

$$\begin{aligned} H_1 &= \frac{1}{4}(1-\xi)^2(2+\xi) \\ H_2 &= \frac{1}{4}(1-\xi)^2(\xi+1) \\ H_3 &= \frac{1}{4}(1+\xi)^2(2-\xi) \\ H_2 &= \frac{1}{4}(1+\xi)^2(\xi-1) \end{aligned} \tag{B.12}$$

The Hermitian shape functions can now be used to write v in the form

$$v(\xi) = H_1v_1 + H_2\left(\frac{dv}{d\xi}\right)_1 + H_3v_2 + H_4\left(\frac{dv}{d\xi}\right)_2 \tag{B.13}$$

The coordinates transform by the relationship

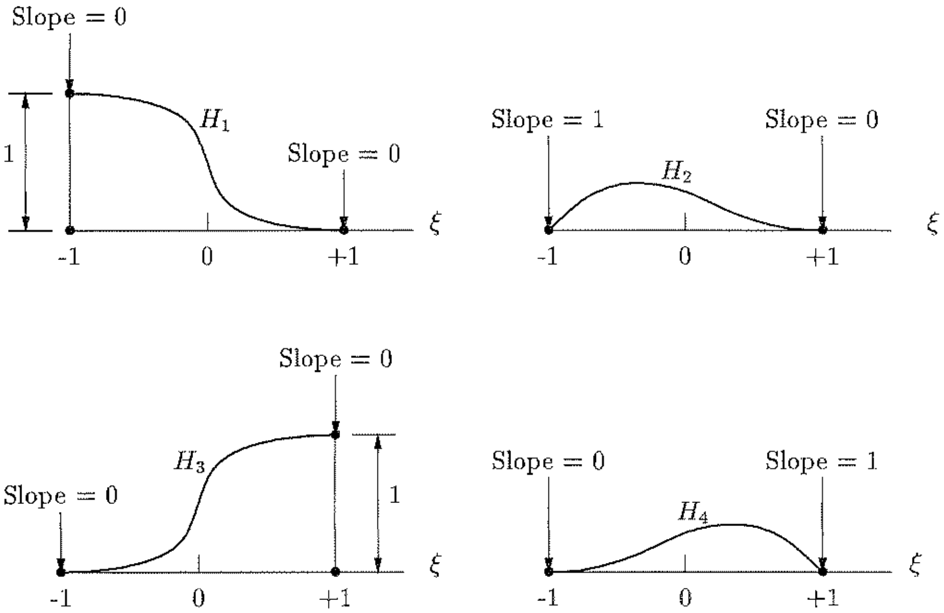


Figure B.4: Hermitian shape functions.

$$x = \frac{1-\xi}{2}x_1 + \frac{1+\xi}{2}x = \frac{x_1+x_2}{2} + \frac{x_2-x_1}{2}\xi \tag{B.14}$$

And, since $l_e = x_2 - x_1$ is the length of the element, we get

$$dx = \frac{l_e}{2}d\xi \tag{B.15}$$

The chain rule $dv/d\xi = (dv/dx)(dx/d\xi)$ yields

$$\frac{dv}{d\xi} = \frac{l_e}{2}\frac{dv}{dx} \tag{B.16}$$

Noting that dv/dx evaluated at nodes 1 and 2 is q_2 and q_4 respectively, we have

$$v(\xi) = H_1q_1 + \frac{l_e}{2}H_2q_2 + H_3q_3 + \frac{l_e}{2}H_4q_4 \tag{B.17}$$

which, in matrix notation, may be denoted as

$$v = \boldsymbol{Hq} \tag{B.18}$$

with

$$\boldsymbol{H} = \left[H_1 \quad \frac{l_e}{2}H_2 \quad H_3 \quad \frac{l_2}{2}H_4\right] \tag{B.19}$$

Now, from (B.16) it is clear that

$$\frac{dv}{dx} = \frac{2}{l_e}\frac{dv}{d\xi} \quad \text{and} \quad \frac{d^2v}{dx^2} = \frac{4}{l_e^2}\frac{d^2v}{d\xi^2} \tag{B.20}$$

Meanwhile, substituting $v = \boldsymbol{Hq}$, we obtain

$$\left(\frac{d^2\boldsymbol{H}}{d\xi^2}\right) = \begin{bmatrix} \frac{3}{2}\xi & \frac{-1+3\xi}{2}\frac{l_e}{2} & -\frac{3}{2}\xi & \frac{1+3\xi}{2}\frac{l_e}{2} \end{bmatrix} \tag{B.21}$$

In the development based on Galerkin's approach (B.9), we note that

$$EI\frac{d^2\phi}{dx^2}\frac{d^2v}{dx^2} = \boldsymbol{\psi}^T EI\frac{16}{l_e^4}\left(\frac{d^2\boldsymbol{H}}{d\xi^2}\right)^T\left(\frac{d^2\boldsymbol{H}}{d\xi^2}\right)\boldsymbol{q} \tag{B.22}$$

where

$$\boldsymbol{\psi} = [\psi_1 \quad \psi_2 \quad \psi_3 \quad \psi_4]^T \tag{B.23}$$

is the set of generalized virtual displacements on the element, $v = \boldsymbol{Hq}$ and $\phi = \boldsymbol{H\psi}$. Equation B.22 yields the stiffness matrix

$$\boldsymbol{k}_e = \frac{EI}{l_e^3}\begin{bmatrix} 12 & 6l_e & -12 & 6l_e \\ 6l_e & 4l_e^2 & -6l_e & 2l_e^2 \\ -12 & -6l_e & 12 & -6l_e \\ 6l_e & 2l_e^2 & -6l_e & 4l_e^2 \end{bmatrix} \tag{B.24}$$

upon integration, with $\boldsymbol{\psi}^T\boldsymbol{k}_e\boldsymbol{q}$ being the internal virtual work in an element.

B.3 Plane frame element

Plane frame elements are similar to beam elements, except that axial loads and deformations are present. Figure B.5 depicts a frame element in a general orientation. There are two displacement and one rotational degree of freedom at each node. The nodal displacement vector is thus given by

$$\boldsymbol{q} = [q_1 \quad q_2 \quad q_3 \quad q_4 \quad q_5 \quad q_6]^T \tag{B.25}$$

A local coordinate system (x', y') is defined such that x' is orientated along 1-2, with direction cosines l, m (with $l = \cos\theta$, $m = \sin\theta$). Now, the displacement vector in the local coordinate system is

$$\boldsymbol{q}' = [q_1' \quad q_2' \quad q_3' \quad q_4' \quad q_5' \quad q_6']^T \tag{B.26}$$

where q_1' and q_2' are the translations at node 1 and q_3' is the rotation at node 1 in the local coordinate system, as depicted in Figure B.5. Similarly, q_4' and q_5' are the translation of node 2 and q_6' the rotation at node 2.

Recognizing that $q'_3 = q_3$ and that $q'_6 = q_6$, we obtain the local-global transformation

$$\boldsymbol{q}' = \boldsymbol{L}\boldsymbol{q} \tag{B.27}$$

where

$$\boldsymbol{L} = \begin{bmatrix} l & m & 0 & 0 & 0 & 0 \\ -m & l & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & l & m & 0 \\ 0 & 0 & 0 & -m & l & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{B.28}$$

q'_2, q'_3, q'_5 and q'_6 are the beam degrees of freedom, while q'_1 and q'_4 are similar to the displacements of a rod element. Superimposing the two stiffness matrices, the following element stiffness matrix is obtained for a frame element

$$\boldsymbol{k}'_e = \begin{bmatrix} \frac{EA}{l_e} & 0 & 0 & \frac{-EA}{l_e} & 0 & 0 \\ & \frac{12EI}{l_e^3} & \frac{6EI}{l_e^2} & 0 & \frac{-12EI}{l_e^3} & \frac{6EI}{l_e^2} \\ & & \frac{4EI}{l_e} & 0 & \frac{-6EI}{l_e^2} & \frac{2EI}{l_e} \\ & & & \frac{EA}{l_e} & 0 & 0 \\ & & & & \frac{12EI}{l_e^3} & \frac{-6EI}{l_e^2} \\ symm & & & & & \frac{4EI}{l_e} \end{bmatrix} \tag{B.29}$$

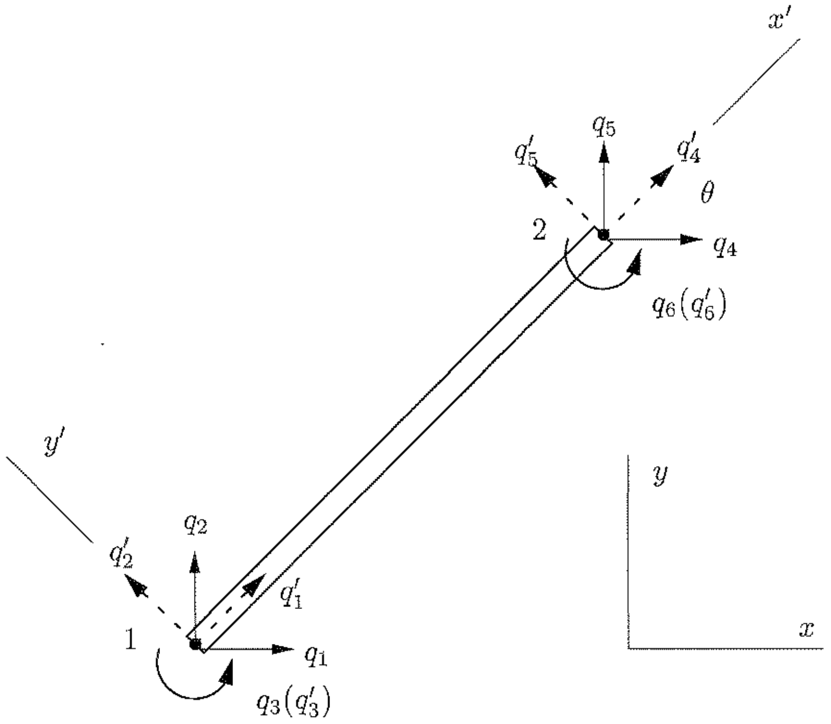


Figure B.5: Plane frame element.

From Galerkin's approach, we recognize that the internal virtual work of an element is

$$W_e = \boldsymbol{\psi}'^T \boldsymbol{k}'_e \boldsymbol{q}' = \boldsymbol{\psi}^T \boldsymbol{L}^T \boldsymbol{k}'_e \boldsymbol{L}\boldsymbol{q} \tag{B.30}$$

where $\boldsymbol{\psi}'$ and $\boldsymbol{\psi}$ are the virtual nodal displacements in the local and global coordinate systems respectively. From (B.30) we recognize the element stiffness matrix in the global coordinate system to be

$$\boldsymbol{k}_e = \boldsymbol{L}^T \boldsymbol{k}'_e \boldsymbol{L} \tag{B.31}$$

B.4 Three dimensional frame element

Figure B.6 shows a general three dimensional frame element. The nodal displacement vector in the local coordinate system is thus given by

$$\boldsymbol{q}' = [q'_1 \;\; q'_2 \;\; q'_3 \;\; q'_4 \;\; q'_5 \;\; q'_6 \;\; q'_7 \;\; q'_8 \;\; q'_9 \;\; q'_{10} \;\; q'_{11} \;\; q'_{12}]^T \tag{B.32}$$

where q'_1, q'_2 and q'_3 are the translations at node 1 along x', y' and z' respectively and q'_4, q'_5 and q'_6 are the rotations at node 1.

Orientation of the local $x' - y' - z'$ coordinate system is established with the use of three points. Point 1 and 2 are the element end points, and the $x'-$ axis is along the line joining 1 and 2. The y' axis lies within the plane defined by points 1, 2 and 3, as shown in Figure B.6.

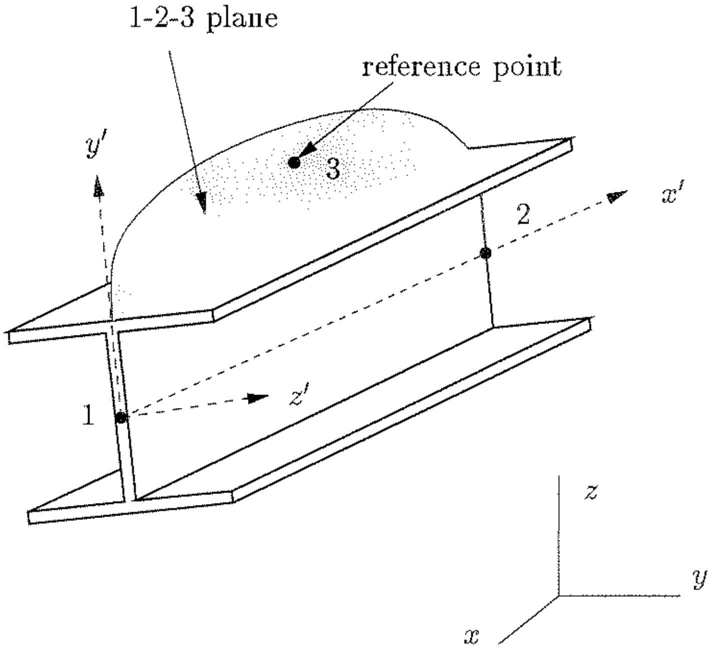


Figure B.6: Three dimensional frame element.

The (12×12) element stiffness matrix $\boldsymbol{k}'$ in the local coordinate system is obtained by a straightforward generalization of (B.29) as

$$\boldsymbol{k}' = \left[\begin{array}{cccccccccccc}
AS & 0 & 0 & 0 & 0 & 0 & -AS & 0 & 0 & 0 & 0 & 0 \\
 & a_{z'} & 0 & 0 & 0 & b_{z'} & 0 & -a_{z'} & 0 & 0 & 0 & b_{z'} \\
 & & a_{y'} & 0 & -b_{y'} & 0 & 0 & 0 & -a_{y'} & 0 & -b_{y} & 0 \\
 & & & TS & 0 & 0 & 0 & 0 & 0 & -TS & 0 & 0 \\
 & & & & c_{y'} & 0 & 0 & 0 & b_{y'} & 0 & d_{y'} & 0 \\
 & & & & & c_{z'} & 0 & -b_{z'} & 0 & 0 & 0 & d_{z'} \\
 & & & & & & AS & 0 & 0 & 0 & 0 & 0 \\
 & & & & & & & a_{z'} & 0 & 0 & 0 & -b_{z'} \\
 & & & & & & & & c_{y'} & 0 & b_{y'} & 0 \\
 & & & & & & & & & TS & 0 & 0 \\
 & & & & & & & & & & c_{y'} & 0 \\
Symm & & & & & & & & & & & c_{z'}
\end{array}\right] \qquad \text{(B.33)}$$

where $AS = AE/L$, $L =$ element length, $TS = GJ/L$, $a_{z'} = 12EI_{z'}/L^3$, $b_{z'} = 6EI_{z'}/L^2$, $c_{z'} = 4EI_{z'}/L$, $d_{z'} = 2EI_{z'}/L$, $a_{y'} = 12EI_{y'}/L^3$, etc. The global-local transformation is given by

$$\boldsymbol{q}' = \boldsymbol{L}\boldsymbol{q} \qquad \text{(B.34)}$$

With the (12×12) transformation matrix $\boldsymbol{L}$ defined as

$$\boldsymbol{L} = \begin{bmatrix} \boldsymbol{\lambda} & & & \mathbf{0} \\ & \boldsymbol{\lambda} & & \\ & & \boldsymbol{\lambda} & \\ \mathbf{0} & & & \boldsymbol{\lambda} \end{bmatrix} \qquad \text{(B.35)}$$

where $\boldsymbol{\lambda}$ is the (3×3) matrix of direction cosines:

$$\boldsymbol{\lambda} = \begin{bmatrix} l_1 & m_1 & n_1 \\ l_2 & m_2 & n_2 \\ l_3 & m_3 & n_3 \end{bmatrix} \qquad \text{(B.36)}$$

Here, l_1, m_1 and n_1 are the cosines of the angles between the $x'-$axis and the global $x-$, $y-$ and $z-$axes respectively. Similarly, l_2, m_2 and n_2 are the cosines of the angles between the $y'-$axis and the $x-$, $y-$ and $z-$axes and l_3, m_3 and n_3 are associated with the $z-$axis.

Finally, the element stiffness in the global coordinate system is given by

$$\boldsymbol{k} = \boldsymbol{L}^T\boldsymbol{k}'\boldsymbol{L} \qquad \text{(B.37)}$$

where the $\boldsymbol{k}'$ matrix is given by (B.33).

Appendix C

Flat shell finite element

In this appendix the formulation of flat shells as an assembly of membrane and plate elements is discussed. This discussion is taken mainly from Groenwold [12, 13], who implemented flat shell finite elements with in-plane drilling degrees of freedom in the finite element (FE) program EDSAP [46]. The drilling dof allow for beam-shell interactions, and are based on the continuum mechanics definition of rotation.

C.1 A General Flat Shell Formulation

It is well known that shells are amongst the most complex structures to analyze numerically. A number of shell theories exist, i.e. deep, shallow or flat shell theory, each associated with particular assumptions and difficulties.

In this section the formulation of flat shell elements as an assembly of membrane and plate elements is briefly discussed. Further detail may for instance be found in the standard work of Zienkiewicz and Taylor [49].

This discussion is restricted to the assembly of the modified QCnD membrane element of Ibrahimbegovic *et al.* [11] and the Bathe-Dvorkin [48] assumed strain plate element. The resultant shell element is denoted QCnD-SA, where n indicates the number of integration points employed in the membrane component. However, the formulation is easily generalized to incorporate different 4-node membrane or plate elements. Furthermore, while drilling degrees of freedom are included in the following formulation, the presence of drilling degrees of freedom is not an absolute requirement. Alternative approaches can work in locally defined systems with 5 degrees of freedom per node. Small artificial stiffnesses associated with the in-plane degrees of freedom can then be incorporated. These artificial stiffnesses are modeled with an equivalent "imaginary" torsional spring at each node.

Nevertheless, drilling degrees of freedom are attractive in the formulation, since their presence allows for beam-shell connections and enriches the calculated displacement field. Hence they enhance the element accuracy. Finally, the drilling dof are based on the continuum mechanics definition of rotation, which does not require the inclusion of imaginary in-plane torsional stiffness.

The element force-displacement relationship of the QCnD membrane elements is defined in local membrane coordinates as

$$[\boldsymbol{K}+\boldsymbol{P}]\boldsymbol{q}_m \;=\; \boldsymbol{f}_m \tag{C.1}$$

where $\boldsymbol{K}+\boldsymbol{P}$ denotes the stiffness matrix, $\boldsymbol{q}_m$ the element displacements and $\boldsymbol{f}_m$ the element loads. $\boldsymbol{P}$ represents a penalty stiffness which relates the in-plane translations and rotations, and prevents mechanisms. In partitioned form, Equation (C.1) may be re-constructed as

$$\begin{bmatrix} (\boldsymbol{K}_m+\boldsymbol{P}) & \boldsymbol{K}_{mh} \\ \boldsymbol{K}_{mh}^T & \boldsymbol{K}_h \end{bmatrix}\boldsymbol{q}_m \;=\; \boldsymbol{f}_m \tag{C.2}$$

where subscript m indicates those terms associated with the 4 membrane corner nodes only. Subscript h indicates those terms associated with the hierarchical membrane bubble shape function (i.e. the center node), while $\boldsymbol{K}_{mh}$ indicates the coupling stiffness between these two. The terms arising due to $\boldsymbol{K}_h$ are eliminated on the element level using static condensation.

The unknown nodal displacements $\boldsymbol{q}_m$ and the specified consistent nodal loads $\boldsymbol{f}_m$ are defined by

$$\boldsymbol{q}_m^i \;=\; [u_i \;\; v_i \;\; \theta_{zi}]^T \tag{C.3}$$

$$\boldsymbol{f}_m^i \;=\; [U_i \;\; V_i \;\; M_{zi}]^T \tag{C.4}$$

where θ_{zi} is the in-plane nodal rotation and M_{zi} the in-plane nodal moment.

Similarly, the Mindlin-Reissner plate force-displacement relationship is written as

$$(\boldsymbol{K}_b+\bar{\boldsymbol{K}}_s)\boldsymbol{q}_p \;=\; \boldsymbol{f}_p \tag{C.5}$$

where $\bar{\boldsymbol{K}}_s$ is the shear stiffness associated with the substitute assumed strain interpolation [48], to prevent shear locking. Subscripts b, s and p denote bending, shear and plate respectively. The displacements $\boldsymbol{q}_p$ and the consistent nodal loads $\boldsymbol{f}_p$ are respectively defined by

$$\boldsymbol{q}_p^i \;=\; [w_i \;\; \theta_{xi} \;\; \theta_{yi}]^T \tag{C.6}$$

$$\boldsymbol{f}_p^i \;=\; [W_i \;\; M_{xi} \;\; M_{yi}]^T \tag{C.7}$$

The shell element stiffness matrix $\boldsymbol{K}$ therefore reduces to

$$\boldsymbol{K} \;=\; \begin{bmatrix} (\boldsymbol{K}_m+\boldsymbol{P}) & 0 & \boldsymbol{K}_{mh} \\ & (\boldsymbol{K}_b+\bar{\boldsymbol{K}}_s) & 0 \\ symm & & \boldsymbol{K}_h \end{bmatrix} \tag{C.8}$$

The local shell force-displacement is

$$\boldsymbol{K}\boldsymbol{q} \;=\; \boldsymbol{f} \tag{C.9}$$

where the shell nodal displacements and loads respectively are

$$\boldsymbol{q}_i = [u_i \;\; v_i \;\; w_i \;\; \theta_{xi} \;\; \theta_{yi} \;\; \theta_{zi}]^T \tag{C.10}$$

and

$$\boldsymbol{f}_i = [U_i \;\; V_i \;\; W_i \;\; M_{xi} \;\; M_{yi} \;\; M_{zi}]^T \tag{C.11}$$

In Equation (C.8), the partition stiffness matrices are given by

$$\boldsymbol{K}_m = \int_\Omega [\boldsymbol{B}_m \, \boldsymbol{G}]^T \, \boldsymbol{C}^m \, [\boldsymbol{B}_m \, \boldsymbol{G}] \tag{C.12}$$

where $\boldsymbol{B}_m$ is the standard bi-linear strain-displacement operator and $\boldsymbol{G}$ defines the rotation terms. ($\boldsymbol{P}^e$ is a penalty stiffness, which relates the in-plane displacements and rotations.)

$$\boldsymbol{K}_b = \int_\Omega \boldsymbol{B}_b^T \boldsymbol{C}^b \boldsymbol{B}_b \, \mathrm{d}\Omega \,, \tag{C.13}$$

$$\bar{\boldsymbol{K}}_s = \int_\Omega \bar{\boldsymbol{B}}_s^T \boldsymbol{C}^s \bar{\boldsymbol{B}}_s \, \mathrm{d}\Omega \tag{C.14}$$

$$\boldsymbol{K}_h = \int_\Omega \boldsymbol{B}_h^T \boldsymbol{C}^m \boldsymbol{B}_h \, \mathrm{d}\Omega \tag{C.15}$$

and

$$\boldsymbol{K}_{mh} = \int_\Omega [\boldsymbol{B}_m \, \boldsymbol{G}]^T \, \boldsymbol{C}^m \, \boldsymbol{B}_h \, \mathrm{d}\Omega \tag{C.16}$$

The operators in (C.12) through (C.16) are explicitally given in [12, 13].

C.2 Constitutive Relationship

In this section the constitutive relationship for isotropic materials is presented. For isotropic material and plane stress, the constitutive modulus $\boldsymbol{C}^m$ has the form

$$\boldsymbol{C}^m = \frac{E}{(1-\nu^2)} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & (1-\nu)/2 \end{bmatrix} \tag{C.17}$$

and where E and ν are Young's modulus and Poisson's ratio, respectively. Furthermore,

$$\boldsymbol{C}^b = \frac{Et^3}{12(1-\nu^2)} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & (1-\nu)/2 \end{bmatrix} \tag{C.18}$$

where t is the element thickness, and

$$\boldsymbol{C}^s = Gt \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{C.19}$$

where G is the shear modulus.

C.3 Local-Global Transformations

The element stiffness matrix described by Equations (C.8) is formed in a local element coordinate system. Element assembly is performed in the global coordinate system, in which system the nodal coordinates, forces and tractions are also prescribed. The relation between the local element stiffness $\boldsymbol{K}^e$ and the global element stiffness $\boldsymbol{K}$ is given by:

$$\boldsymbol{K} = \boldsymbol{T}^T \boldsymbol{K}^e \boldsymbol{T}^T \tag{C.20}$$

where $\boldsymbol{T}$ is the applicable rotation transformation matrix (e.g. see [49]).

Appendix D

LFOPC

The LFOPC (Leap-Frog Optimization Program for Constrained optimization) algorithm applies a dynamic trajectory method for unconstrained optimization, originally proposed and developed by Snyman [55, 56], to a penalty function formulation of the constrained problem [15, 16]. The original dynamic trajectory method is based on the physical model of the motion of a particle of unit mass in an n-dimensional conservative force field, where the potential energy of the particle is given by the function to be minimized $F(\boldsymbol{x})$. This method is a proven reliable and robust method.

The LFOPC algorithm is well suited to the type of optimization problems encountered in the engineering field, since engineering problems often consist of:

1. functions (cost function or constraint function) which are expensive to evaluate,
2. have noise present in these functions, which can originate from experimental data, or from numerical inaccuracies,
3. functions which may be discontinuous or non-differentiable,
4. functions with many local minima,
5. functions with regions in the design space where the function is badly scaled or undefined and
6. functions with a large number of design variables.

These difficulties limit the application of traditional optimization algorithms to engineering problems.

D.1 Basic dynamic model

Consider the general nonlinear optimization problem:

$$\begin{aligned}
&\min_{\boldsymbol{x}} F(\boldsymbol{x}), \qquad \boldsymbol{x} = (x_1, x_2, \ldots, x_n) \in \Re^n \\
&\text{subject to } g_j(\boldsymbol{x}) \leq 0, \qquad j = 1, 2, \ldots, p \\
&\text{and } h_k(\boldsymbol{x}) = 0, \qquad k = 1, 2, \ldots, q
\end{aligned} \tag{D.1}$$

Assume a particle of unit mass in an n-dimensional conservative force field with potential energy at $\boldsymbol{x}$ given by $F(\boldsymbol{x})$, then at $\boldsymbol{x}$ the force on the particle is given by:

$$\boldsymbol{a} = \ddot{\boldsymbol{x}} = -\nabla F(\boldsymbol{x}) \tag{D.2}$$

from which it follows that for the time interval $[0, t]$:

$$\frac{1}{2}||\dot{\boldsymbol{x}}(t)||^2 - \frac{1}{2}||\dot{\boldsymbol{x}}(0)||^2 = F(\boldsymbol{x}(0)) - F(\boldsymbol{x}(t)) \tag{D.3}$$

$$T(t) - T(0) = F(0) - F(t); \text{ or } F(t) + T(t) = F(0) + T(0) = K \tag{D.4}$$

where F(t) and T(t) are the potential energy and kinetic energy of the particle respectively at time t and K is a constant.

It may be noted that $\Delta F = -\Delta T$, and therefore as long as T increases, F decreases. This forms the basis of the dynamic trajectory method.

D.2 Basic LFOP algorithm for unconstrained optimization

Given F($\boldsymbol{x}$) and the starting point $\boldsymbol{x}(0) = \boldsymbol{x}^0$, compute the trajectory by solving the initial value problem (IVP):

$$\begin{aligned}
&\ddot{\boldsymbol{x}}(t) = -\nabla F(\boldsymbol{x}(t)) \\
&\dot{\boldsymbol{x}}(0) = \boldsymbol{0} \\
&\boldsymbol{x}(0) = \boldsymbol{x}^0
\end{aligned} \tag{D.5}$$

Monitor $\dot{\boldsymbol{x}}(t) = \boldsymbol{v}(t)$, since while $T(t) = \frac{1}{2}||\boldsymbol{v}||^2$ increases, $F(\boldsymbol{x}(t))$ decreases, and the solution $\boldsymbol{x}$ moves towards the minimum of F. When $||\boldsymbol{v}(t)||$ decreases, the solution $\boldsymbol{x}$ is moving "uphill". In this case an interfering strategy is applied to extract energy from the particle, in order to increase the likelihood of descent.

In practice, the numerical integration of the IVP in (D.5) is performed by means of the leap-frog method:

Compute for $k = 0, 1, \ldots$ and time step Δt:

$$\begin{aligned} \boldsymbol{x}^{(k+1)} &= \boldsymbol{x}^{(k)} + \boldsymbol{v}^{k}\Delta t \\ \boldsymbol{v}^{(k+1)} &= \boldsymbol{v}^{(k)} + \boldsymbol{a}^{k+1}\Delta t \end{aligned} \tag{D.6}$$

where

$$\begin{aligned} \boldsymbol{a}^{(k)} &= -\nabla F(\boldsymbol{x}^{(k)}) \\ \boldsymbol{v}^{(0)} &= \frac{1}{2}\boldsymbol{a}^{(0)}\Delta t \end{aligned} \tag{D.7}$$

A typical interfering strategy is

$$\begin{aligned} &\text{If } ||\boldsymbol{v}^{(k+1)}|| \geq ||\boldsymbol{v}^{(k)}|| \quad \text{continue} \\ &\text{otherwise set } \boldsymbol{v}^{(k)} = \frac{\boldsymbol{v}^{(k+1)} + \boldsymbol{v}^{(k)}}{4};\ \boldsymbol{x}^{(k)} = \frac{\boldsymbol{x}^{(k+1)} + \boldsymbol{x}^{(k)}}{2} \\ &\qquad \text{compute the new } \boldsymbol{v}^{(k+1)} \text{ and continue} \end{aligned} \tag{D.8}$$

The following three termination criteria are used:

1. Stop if $||\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k+1)}|| < \varepsilon_x$,
2. Stop if $||\boldsymbol{a}^{(k)}|| < \varepsilon_g$ and
3. Stop if maximum number of iteration is exceeded.

Other heuristics are incorporated, such as specifying a maximum trajectory step size, an empirical formulae for determining the initial value for Δt, and a scheme to magnify and reduce Δt in order to minimize the number of iterations required for convergence.

D.3 Modifications for constraints

Constraints are accommodated in the algorithm by means of a penalty function formulation. This formulation solves the constrained minimization problem stated in (D.1), by unconstrained minimization of a modified cost function, $P(\boldsymbol{x}, \mu)$:

$$P(\boldsymbol{x}, \mu) = F(\boldsymbol{x}) + \mu \sum_{j=1}^{m} [g_j(\boldsymbol{x})]^2 u_j(g_j) + \mu \sum_{k=1}^{p} [h_k(\boldsymbol{x})]^2 \tag{D.9}$$

where

$$u_j(g_j) = \begin{cases} 0 & \text{if } g_j(\boldsymbol{x}) \leq 0 \\ 1 & \text{if } g_j(\boldsymbol{x}) > 0 \end{cases} \tag{D.10}$$

and where $\mu >> 0$ is the overall penalty parameter.

The solution to the minimization problem is performed in three phases:

Phase 0

Given some $\boldsymbol{x}^0$, apply LFOP to $P(\boldsymbol{x}, \mu_0)$ with an overall penalty parameter of $\mu = \mu_0$ to give $\boldsymbol{x}^*(\mu_0)$.

Phase 1

With $\boldsymbol{x}^0 = \boldsymbol{x}^*(\mu_0)$ and $\mu = \mu_1 > \mu_0$, apply LFOP to $P(\boldsymbol{x}, \mu_1)$ to give $\boldsymbol{x}^*(\mu_1)$. Identify the active constraints $i_a = 1, 2, \ldots, n_a$ where $g_{i_a}(\boldsymbol{x}^*(\mu_1)) > 0$.

Phase 2

With $\boldsymbol{x}^0 = \boldsymbol{x}^*(\mu_1)$ apply LFOP to:

$$\text{minimize } P_a(\boldsymbol{x}, \mu_1) = \mu_1 \sum_{k=1}^{p} h_k^2(\boldsymbol{x}) + \mu_1 \sum_{i_a=1}^{n_a} g_{i_a}^2(\boldsymbol{x}) \tag{D.11}$$

to give $\boldsymbol{x}^*$, where $Pa(\boldsymbol{x}, \mu_1)$ is a function involving only the equality constraint h_j, $j = 1, 2, \ldots, p$, and the active constraints g_{i_a}, $i_a = 1, 2, \ldots, n_a$.

Appendix E

Dynamic-Q

The Dynamic-Q algorithm was proposed and developed by Snyman *et al.* [16], and applied to a number of difficult engineering problems [58, 59, 60]. In the field of engineering, the functions in an optimization problem are often very expensive to evaluate. Such problems can be solved economically by using approximation methods (such as Dynamic-Q and SQP) which require fewer function evaluations than standard gradient-based descent algorithms.

Dynamic-Q adopts a successive approximation approach to solve the minimization problem in (E.1). It solves successive subproblems constructed from the original problem (E.1). These subproblems are analytically simple and can easily and economically be solved by the dynamic trajectory method (LFOPC).

E.1 Basic algorithm

Consider the general nonlinear optimization problem:

$$\begin{aligned} &\min_{\boldsymbol{x}} F(\boldsymbol{x}), \qquad \boldsymbol{x} = (x_1, x_2, \ldots, x_n) \in \Re^n \\ &\text{subject to } g_j(\boldsymbol{x}) \leq 0, \qquad j = 1, 2, \ldots, p \\ &\text{and } h_k(\boldsymbol{x}) = 0, \qquad k = 1, 2, \ldots, q \end{aligned} \tag{E.1}$$

E.1.1 Construction of successive sub-problems

In the Dynamic-Q approach, successive subproblems $P[i]$, $i = 1, 2, \ldots$ are generated at successive approximations $\boldsymbol{x}^i$ to the solution $\boldsymbol{x}^*$, by constructing *spherically quadratic* approximations $\tilde{F}(\boldsymbol{x})$, $\tilde{g}_j(\boldsymbol{x})$ and $\tilde{h}_k(\boldsymbol{x})$ to $F(\boldsymbol{x})$, $g_j(\boldsymbol{x})$ and $h_k(\boldsymbol{x})$. These approximation functions, evaluated at a point $\boldsymbol{x}^i$, are given by

$$
\begin{aligned}
\tilde{F}(\boldsymbol{x}) &= F(\boldsymbol{x}^i) + \nabla^T F(\boldsymbol{x}^i)(\boldsymbol{x}-\boldsymbol{x}^i) + \frac{1}{2}(\boldsymbol{x}-\boldsymbol{x}^i)^T \boldsymbol{A}(\boldsymbol{x}-\boldsymbol{x}^i) & \\
\tilde{g}_j(\boldsymbol{x}) &= g_j(\boldsymbol{x}^i) + \nabla^T g_j(\boldsymbol{x}^i)(\boldsymbol{x}-\boldsymbol{x}^i) + \frac{1}{2}(\boldsymbol{x}-\boldsymbol{x}^i)^T \boldsymbol{B}_j(\boldsymbol{x}-\boldsymbol{x}^i), & j = 1,2,\dots,p \\
\tilde{h}_k(\boldsymbol{x}) &= h_k(\boldsymbol{x}^i) + \nabla^T h_k(\boldsymbol{x}^i)(\boldsymbol{x}-\boldsymbol{x}^i) + \frac{1}{2}(\boldsymbol{x}-\boldsymbol{x}^i)^T \boldsymbol{C}_k(\boldsymbol{x}-\boldsymbol{x}^i), & k = 1,2,\dots,q
\end{aligned}
\tag{E.2}
$$

with the Hessian matrices $\boldsymbol{A}$, $\boldsymbol{B}_j$ and $\boldsymbol{C}_k$ taking on the simple forms

$$
\boldsymbol{A} = \mathrm{diag}(a, a, \dots, a) = a\boldsymbol{I}, \qquad \boldsymbol{B}_j = b_j\boldsymbol{I}, \qquad \boldsymbol{C}_k = c_k\boldsymbol{I} \tag{E.3}
$$

Clearly, the identical entries along the diagonal of the Hessian matrices indicate that the approximate subproblems $P[i]$ are indeed spherically quadratic.

For the first subproblem ($i = 0$) a linear approximation is formed by setting the curvatures a, b_j and c_k to zero. Thereafter a, b_j and c_k are chosen so that the approximating functions (E.2) interpolate their corresponding actual functions at both $\boldsymbol{x}^i$ and $\boldsymbol{x}^{i-1}$. These conditions imply that for $i = 1, 2, \dots$

$$
\begin{aligned}
a &= \frac{2[F(\boldsymbol{x}^{i-1}) - F(\boldsymbol{x}^i) - \nabla^T F(\boldsymbol{x}^i)(\boldsymbol{x}^{i-1}-\boldsymbol{x}^i)]}{||\boldsymbol{x}^{i-1}-\boldsymbol{x}^i||^2} \\
b_j &= \frac{2[g_j(\boldsymbol{x}^{i-1}) - g_j(\boldsymbol{x}^i) - \nabla^T g_j(\boldsymbol{x}^i)(\boldsymbol{x}^{i-1}-\boldsymbol{x}^i)]}{||\boldsymbol{x}^{i-1}-\boldsymbol{x}^i||^2} \\
c_k &= \frac{2[h_k(\boldsymbol{x}^{i-1}) - h_k(\boldsymbol{x}^i) - \nabla^T h_k(\boldsymbol{x}^i)(\boldsymbol{x}^{i-1}-\boldsymbol{x}^i)]}{||\boldsymbol{x}^{i-1}-\boldsymbol{x}^i||^2}
\end{aligned}
\tag{E.4}
$$

If the gradient vectors ∇F, ∇g_j and ∇h_k are not known analytically, they may be approximated from functional data by means of first-order forward finite differences.

The particular choice of spherically quadratic approximations in the Dynamic-Q algorithm has implications on the computational and storage requirements of the method. Since the second derivatives of the objective function and constraints are approximated using function and gradient data, the $O(n^2)$ calculations and storage locations, which would usually be required for these second derivatives, are not needed. The computational and storage resources for the Dynamic-Q method are thus reduced to $O(n)$. At most, $4+p+q+r+s$ n-vectors need to be stored (where p, q, r and s) are respectively the number of inequality and equality constraints and the number of lower and upper bounds on the variables). These savings become significant when the number of variables becomes large. For this reason, the Dynamic-Q method is well suited, for example, to engineering problems where a large number of variables are commonly present.

E.1.2 Variable bounds

In many optimization problems, additional simple side constraints of the form $\hat{k} \leq x_i \leq \check{k}$ occur. Constraints $\hat{k}$ and $\check{k}$ respectively represent lower and upper bounds on variable x_i. Since these constraints are of a simple form (having zero curvature), they need not be approximated in the Dynamic-Q method and are instead explicitly treated as special linear inequality constraints.

Constraints corresponding to lower and upper bounds are respectively of the form

$$\begin{aligned} \hat{g}_l(\boldsymbol{x}) &= \hat{k}_{vl} - x_{vl} \leq 0, \qquad l = 1, 2, \ldots, r \leq n, \quad \text{and} \\ \check{g}_m(\boldsymbol{x}) &= x_{wm} - \check{k}_{wm} \leq 0, \qquad m = 1, 2, \ldots, s \leq n \end{aligned} \tag{E.5}$$

where $vl \in (v1, v2, \ldots, vr)$ the set of r subscripts corresponding to the set of variables for which respective lower bounds $\hat{k}_{vl}$ are prescribed, and $wm \in (w1, w2, \ldots, ws)$ the set of s subscripts corresponding to the set of variables for which respective upper bounds $\check{k}_{wm}$ are prescribed. The subscripts vl and wm are used since there w1ill, in general, not be n lower and upper bounds, i.e. usually $r \neq n$ and $s \neq n$.

E.1.3 Move limits

In order to obtain convergence to the solution in a controlled and stable manner, move limits are placed on the variables. For each approximate subproblem $P[i]$ the move limit takes the form of an additional single inequality constraint

$$g_\delta = ||\boldsymbol{x} - \boldsymbol{x}^{i-1}||^2 - \delta^2 \leq 0 \tag{E.6}$$

where δ is an appropriately chosen step limit and $\boldsymbol{x}^{i-1}$ is the solution to the previous subproblem.

The approximate subproblem, constructed at $\boldsymbol{x}^i$, to the optimization problem (E.1) (plus simple side constraints (E.5) and move limit (E.6)), thus becomes $P[i]$:

$$\begin{aligned} &\min_{\boldsymbol{x}} \tilde{F}(\boldsymbol{x}), \qquad \boldsymbol{x} = (x_1, x_2, \ldots, x_n) \in \Re^n \\ &\text{subject to } \tilde{g}_j(\boldsymbol{x}) \leq 0, \qquad j = 1, 2, \ldots, p \\ &\tilde{h}_k(\boldsymbol{x}) \leq 0, \qquad k = 1, 2, \ldots, q \\ &\hat{g}_l(\boldsymbol{x}) \leq 0, \qquad l = 1, 2, \ldots, r \\ &\check{g}_m(\boldsymbol{x}) \leq 0, \qquad m = 1, 2, \ldots, s \\ &g_\delta(\boldsymbol{x}) = ||\boldsymbol{x} - \boldsymbol{x}^{i-1}||^2 - \delta^2 \leq 0 \end{aligned} \tag{E.7}$$

with solution $\boldsymbol{x}^{*i}$.

In the Dynamic-Q method the subproblems are solved using the dynamic trajectory method, or "leap-frog" method of Snyman (see Appendix D) for unconstrained optimization applied to a penalty function formulation.

E.2 Formal Dynamic-Q procedure

The Dynamic-Q procedure may be written as follows:

1. Choose a starting point $\boldsymbol{x}^0$ and step limit δ. Set $i = 0$.
2. Evaluate $F(\boldsymbol{x}^i)$, $g_j(\boldsymbol{x}^i)$ and $h_k(\boldsymbol{x}^i)$ as well as $\nabla F(\boldsymbol{x}^i)$, $\nabla g_j(\boldsymbol{x}^i)$ and $\nabla h_k(\boldsymbol{x}^i)$. If termination criteria are satisfied then stop.
3. Construct a local approximation $P[i]$ to the optimization problem at $\boldsymbol{x}^i$ using expressions (E.2), (E.3) and (E.4).
4. Solve the approximated subproblem $P[i]$ (E.7) using the constrained optimizer LFOPC with $\boldsymbol{x}^0 = \boldsymbol{x}^i$ (see Appendix D) to give $\boldsymbol{x}^{*i}$.
5. Set $i = i + 1$ and $\boldsymbol{x}^i = \boldsymbol{x}^{*(i-1)}$ and return to step 2.

Appendix F

The genetic algorithm

F.1 Introduction

This discussion is taken mostly from Goldberg [65] and Michaelwicz [66], as well as the notes written by Houck *et al.* [67] for a Matlab implementation of a genetic algorithm.

Algorithms for function optimization are generally limited to convex regular functions. However, many functions are multi-modal, discontinuous or nondifferentiable. Stochastic sampling methods have been used to optimize these functions.

Whereas traditional search techniques use characteristics of the problem to determine the next sampling points (eg. gradients, Hessian, linearity and continuity) stochastic search techniques make no such assumptions. Instead, the next sampled points are determined based on stochastic sampling/decision rules rather than a set of deterministic decision rules.

Genetic algorithms have been used to solve difficult problems which do not posses "nice" properties such as continuity, differentiability satisfaction of the Lipschitz Condition, etc.

A "survival of the fittest" strategy is employed in order to search the solution space [67]. In general, the fittest individuals of a population tend to reproduce and survive to the next generation, thus improving successive generations. Genetic algorithms have been shown to solve linear and nonlinear problems by exploring promising areas through mutation, crossover and selection operations applied to individuals in the population [66].

Issues which require attention before using a genetic algorithm include: chromosome representation, the creation of the initial population, termination criteria and formulation of the selection function.

Figure F.1 depicts the pseudo-code for a simple genetic algorithm. (Only the most important issues are depicted in this figure.)

F.2 Solution representation

Chromosome representation is necessary to describe the individuals in the population. Each chromosome is made up of a sequence of genes from a certain alphabet. An alphabet can

```
---------------------------------------------------
            Procedure : Genetic Algorithm
---------------------------------------------------
  do
     initialize population of N individuals
     while termination criteria are not satisfied
       do
          calculate fitnesses
          apply selection function
          apply crossover operator
          apply mutation operator
       continue
  continue
---------------------------------------------------
```

Figure F.1: Representation of a simple genetic algorithm.

consist of binary digits (0 and 1), floating point numbers, integers, symbols (i.e. A,B,C,D), matrices, etc. In Holland's [68] original paper, the alphabet was limited to binary digits.

F.3 Selection function

There are several schemes frequently used for the selection process: roulette wheel selection and its extensions, scaling techniques, tournament, elitist models, and ranking methods [65, 66]. Bäck [69] provides a summary of the most popular selection methods available, and only the three most popular of these are considered here.

F.3.1 Proportional selection

Proportional selection is also known as Roulette wheel selection. Individuals are selected randomly with a selection probability equal to the ratio of the fitness of an individual to the sum of the entire population's fitnesses as shown in (F.1).

$$P(\boldsymbol{x}_i) = \frac{f(\boldsymbol{x}_i)}{\sum\limits_{j=1}^{n} f(\boldsymbol{x}_j)} \tag{F.1}$$

where $P(\boldsymbol{x}_i)$ is the probability of selecting individual $\boldsymbol{x}_i$. For obvious reasons, proportional selection can only be used where the fitnesses are to be maximized and all fitnesses are strictly positive. It is a relatively simple task, however, to perform transformations in order to accommodate both negative fitnesses as well as minimization.

A problem with proportional selection is that proportional selection becomes little more than random selection when a larger number of individuals have similar fitnesses. This is frequently the case after a relatively large number of iterations, when most individuals in the population have converged to similar fitness values.

Another problem is that an individual that is much better than all the others in a certain generation will tend to get selected a large number of times, thus driving the population to converge to that solution. This is a problem if such a "good" individual is generated in the initial stages of the run, and especially if it is not close to the global optimum (or best possible design point). Both of these problems can be overcome by scaling the fitness to be between some specific upper and lower bounds [65].

F.3.2 Rank-based selection

Ranking methods only require the evaluation function to map the solution to a partially ordered set, thus allowing minimization and negativity. Ranking methods assign the probability of selection based on the rank of solution i when all solutions are sorted. Individuals that have a higher rank (better individuals) have a greater chance of being selected than individuals with a lower rank.

F.3.3 Tournament selection

Tournament selection, like ranking methods, only require the evaluation function to map solutions to a partially ordered set. However, it does not assign probabilities. Tournament selection works by selecting j individuals randomly from the population to compete in a tournament. The best of the j individuals is selected for survival. In order to determine the winner of the tournament, proportional selection can be performed on the competitors. Furthermore, the bias towards better individuals in the population can be adjusted by changing the number of competitors in the tournament.

F.4 Genetic operators

Genetic operators provide the basic search mechanism of the GA. The genetic operators are used to create new solutions based on existing solutions in the population. There are two general categories of operators, namely crossover and mutation. In most implementations, crossover occurs before mutation. Crossover takes two individuals and produces a number of new "children", while mutation alters one individual to produce a single new solution. Naturally, the implementation of the genetic operators depends on the type of representation used.

F.4.1 Crossover

Crossover is analogous to breeding in nature. The basic principle of crossover is that genetic information from two (or even more) individuals is combined to form a new individual.

Crossover, in the majority of implementations, is the most important of the genetic operators. Two different variable representations are considered here, namely binary and real-valued representations.

Binary representation

Binary crossover was originally proposed by Holland [68] and is intended for use with a binary representation. Binary crossover works by selecting two parents and one (or more) crossover point(s). Data is then copied from before the crossover point of one parent and added to the data after the crossover point of the other parent. An example where two 8 bit parents are crossed after bit 3 to produce two offspring is shown in Figure F.2, where x_i and y_i represent the bits of the parents.

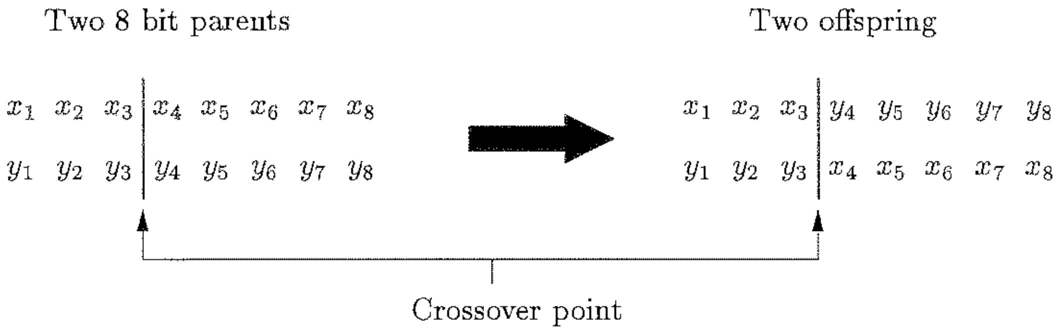

Two 8 bit parents
Two offspring
x_1 x_2 x_3 | x_4 x_5 x_6 x_7 x_8
y_1 y_2 y_3 | y_4 y_5 y_6 y_7 y_8
x_1 x_2 x_3 | y_4 y_5 y_6 y_7 y_8
y_1 y_2 y_3 | x_4 x_5 x_6 x_7 x_8
Crossover point

Figure F.2: An example of the crossover operator with a binary representation.

In this case two offspring are generated for every crossover. However, it is also possible to produce only one child per crossover.

Real-value representation

Simple real valued crossover was introduced by Michaelwicz [66] and is very similar to binary crossover, except that it is applied to real valued representations. The only difference is that the values of x_i and y_i in Figure F.2 are real numbers.

Because simple crossover does not alter the values of the variables, this operator is more limiting. For this reason, Michaelwicz [66] introduced arithmetic crossover. In this improved operator, variables from one parent before the crossover point are copied, the variable at the crossover point is modified using both parents, and then the remaining variables from the other parent are copied. An example of this operator is shown in Figure F.3. The value of a_i is given by

$$a_i = rx_i + (1 - r)y_i \tag{F.2}$$

where r is a random number between 0 and 1.

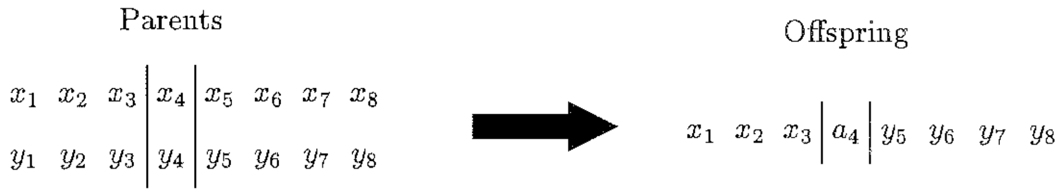

Parents
x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8
y_1 y_2 y_3 y_4 y_5 y_6 y_7 y_8
Offspring
x_1 x_2 x_3 a_4 y_5 y_6 y_7 y_8

Figure F.3: An example of arithmetic crossover.

F.4.2 Mutation

Mutation operators select one individual and then change that individual through manipulation of a few bits to ensure that genetic diversity is maintained in the population. Mutation is a background operator that has a very small effect on the operation of most genetic algorithms, and is applied with a very low probability.

Binary mutation

Binary mutation was originally suggested by Holland [68] as part of a binary genetic algorithm. Binary mutation works by randomly inverting a bit in the binary representation of an individual with a probability p_m as shown in F.3

$$x_i' = \begin{cases} 1 - x_i, & \text{if } r < p_m \\ x_i, & \text{otherwise} \end{cases} \tag{F.3}$$

where r is a random number between 0 and 1.

Real-value mutation

Let $\bar{X}$ and $\bar{Y}$ be two m-dimensional row vectors denoting individuals (parents) from the population. In this case, both $\bar{X}$ and $\bar{Y}$ are real numbers. Furthermore, let a_i and b_i be the lower and upper bound, respectively, for each variable i.

Uniform mutation randomly selects one variable, j and sets it equal to a uniform random number $U \in (a_i, b_i)$:

$$x_i' = \begin{cases} U \in (a_i, b_i), & \text{if } i = j \\ x_i, & \text{otherwise} \end{cases} \tag{F.4}$$

Boundary mutation randomly selects one variable, j, and sets it equal to either its upper or lower bound:

$$x_i' = \begin{cases} a_i, & \text{if } i = j,\ r < 0.5 \\ b_i, & \text{if } i = j,\ r \geq 0.5 \\ x_i, & \text{otherwise} \end{cases} \tag{F.5}$$

where r is a random number between 0 and 1.

Non-uniform mutation randomly selects one variable, j, and sets it equal to a non-uniform random number:

$$x_i' = \begin{cases} x_i + (b_i - x_i)f(G), & \text{if } r_1 < 0.5 \\ x_i - (x_i + a_i)f(G), & \text{if } r_1 \geq 0.5 \\ x_i, & \text{otherwise} \end{cases} \tag{F.6}$$

where:

$f(G) = \left(r_2\left(1 - \frac{G}{G_{max}}\right)\right)^b,$

r_1, r_2 are uniform random numbers between 0 and 1,

G is the current generation,

G_{max} is the maximum number of generations and

b is a shape parameter.

F.5 Initialization and termination

Commonly, a randomly generated initial population is used. However, the initial population can be seeded with potentially good individuals, while the remaining population is generated randomly.

There are numerous possible stopping criteria for optimization algorithms. For GA's, a frequently used stopping criterion is to simply specify a maximum number of generations. This is adequate if the function evaluations are inexpensive. Another termination strategy involves population convergence criteria. There are numerous convergence criteria. When the sum of the deviations among individuals becomes smaller that some specified tolerance, the algorithm can be terminated. Alternatively, the algorithm can be terminated if no improvement in the best known solution is found for a specified number of generations. Furthermore, a target can be set for the best solution, and the algorithm may be terminated if this target is met. These (and other) criteria may be used separately and/or in conjunction with each other.

F.6 The genetic algorithm of Carrol

In this section, the genetic algorithm coded by Carrol [14] is discussed. A brief description of the different functions and operators is presented.

Initialization

In this implementation, a random initial population is generated with no seeding of potentially good solutions.

Representation

A binary representation is used in this implementation. The binary number is mapped to a real number between the specified upper and lower variable bound.

Selection

The selection scheme used is tournament selection with a shuffling technique for choosing random pairs for mating.

Mutation and crossover

The routine includes binary coding for the individuals, jump (uniform) mutation, creep mutation, and the option for single-point or uniform crossover.

Termination

The GA of Carrol simply terminates after a certain specified number of generations. However, in this study an additional criterion is implemented, which ensures that the algorithm terminates after a certain number of generations without improvement in the best function value.

Advanced operators

The implementation of Carrol includes niching (sharing). Niching was originally introduced by Holland [68]. Basically, niching attempts to prevent the *entire* population from converging to a single solution. The "payoff" at a specific design point is shared between the individuals at that specific point. Therefore, the benefit of being at a good design point diminishes as the number of individuals at that design point increases [65].

Niching thus drives the algorithm to explore other promising (possibly better) regions in the design space. The result of niching is that groups of individuals gather around good design points.

Carrol also included elitism in his algorithm. Elitism simply ensures that the best individual is replicated into next generation.

Furthermore, in Carrol's implementation an option to use a micro-GA (μGA) is also included. Basically, a micro-GA utilizes smaller popnlations which converge relatively quickly to a design point. After it is clear that most (or all) of the population is at the same design point, rebirth occurs. Normally elitism is implemented for use in a micro-GA to ensure that best individual is replicated into the next generation, while the remaining individuals are selected randomly during rebirth. A micro-GA, therefore, does not necessarily need to depend on mutation to maintain genetic diversity.

Recommended settings

The following settings are recommended by Carrol [70] for efficient use of his algorithm:

- *Probability of creep mutation* = [(no. chromosomes/no. variables)]/[population size].
- *Probability of jump mutation* = [1]/[population size].
- *Probability of crossover*:
  For single-point crossover a value of 0.6 or 0.7 is recommended.
  For uniform crossover a value of 0.5 is suggested.

Appendix G

Tool force calculations

This appendix describes how the tool force is determined for a specific machining operation. For continuity, the tool forces that were proposed by Smit [51] are used in this study. For this reason the following calculations are taken almost directly from the Masters thesis of Smit [51].

The high frequency, high amplitude forces that may occur during machining operations are not considered here.

In his study, Smit [51] considered only form-milling machining operations as shown in Figure G.1 with a ball-nose cutter as shown in Figure G.2.

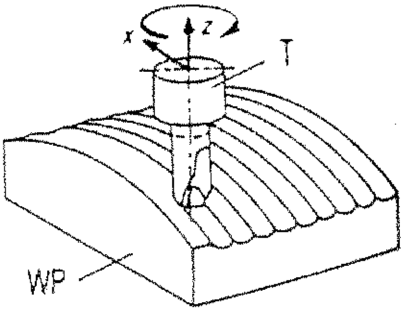


Figure G.1: Form milling.

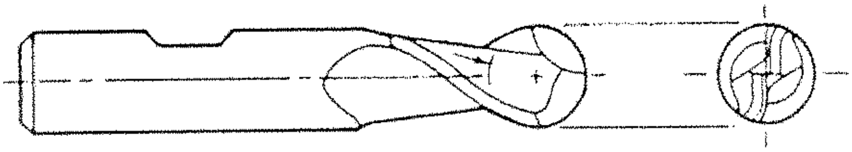


Figure G.2: Ball-nose cutter.

Figure G.3 shows a model for the tool forces on the workpiece. The tool is moving in the x-direction relative to the workpiece. The machining forces on the workpiece due to one tool tooth can be resolved into an active force, F_a, in the work plane and a passive force, F_p, perpendicular to the work plane. the active force, F_a, can be resolved into a force parallel to the movement of the tool tooth, F_c, and a force perpendicular to F_c, which is F_{cN}. F_c and F_{cN} are forces working in on the workpiece. F_c and F_cN are independent of the angle through which the tool has rotated, and is given by Kienzle's machining force equation [71]:

$$F_i = bk_ih^{1-m_i}, \quad i = c, cN, p \tag{G.1}$$

where:

b is the undeformed chip width [mm],
k_i is the specific cutting force [N.mm^{-2}] for an undeformed chip thickness in the range [0.1mm, 1.0mm],
h is the undeformed chip thickness [mm] and
m_i is the incremental value for the workpiece-cutting material pair.

The values of k_i and m_i are empirical and depend on the workpiece material and tool type. These values are available from tables in [71]. The undeformed chip width and chip thickness can be calculated from the feed speed, tool geometry and the tool rotational speed.

The machining operation considered is to form-mill an aluminum workpiece with a two teeth

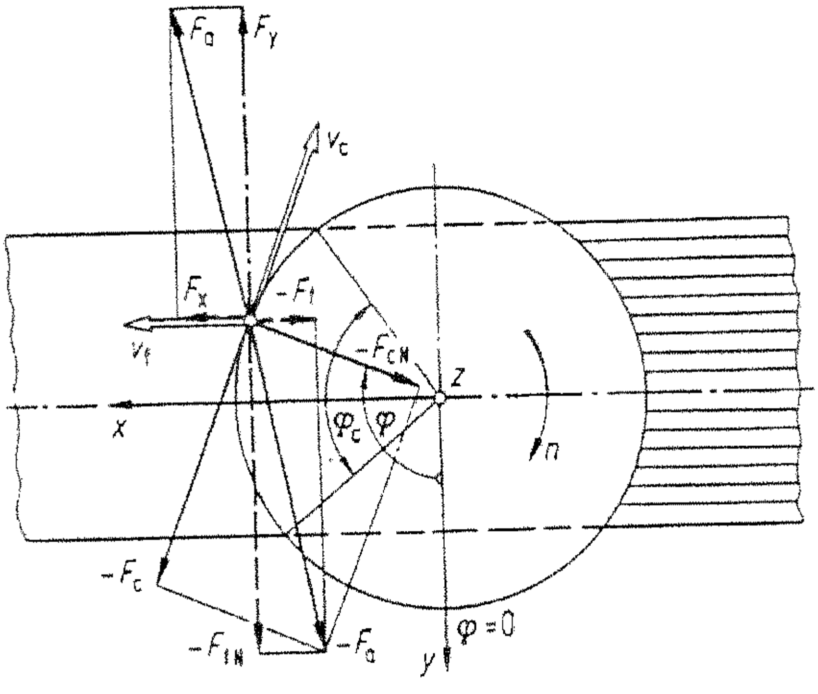


Figure G.3: Model of tool forces.

ball-nose cutter with a shaft diameter, D, of $\frac{1}{2}$ inch. The tool speed is 2000 rpm, the feed rate is 0.25 mm/rev and the cut depth is 6.35 $(D/2)$. The tool forces are thus calculated as $f_x = 1600$ N, $f_y = 400$ N and $f_z = 1800$ N, with the directions depicted in Figure G.4.

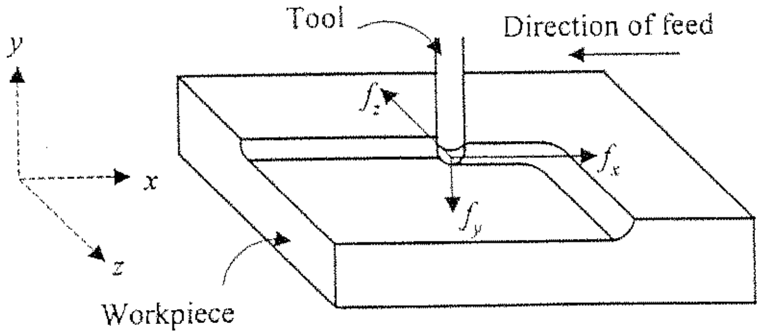


Figure G.4: The machiuing forces acting in on the workpiece.

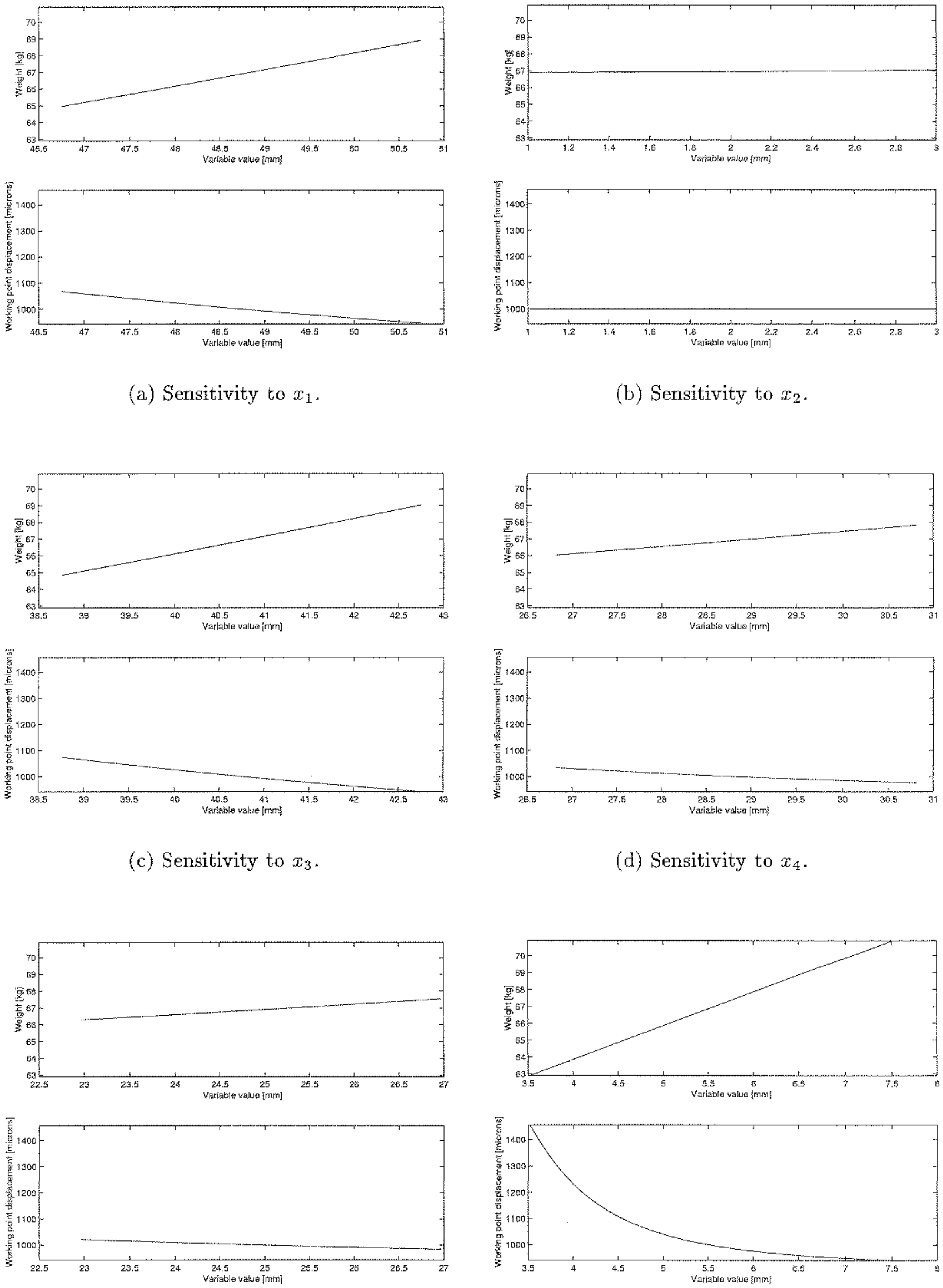
Appendix H

Sensitivity analysis

This appendix contains a sensitivity analysis of the objective and constraint functions to the design variables. These sensitivity analyses are carried out after the optimization process. The *best* design of all the algorithms is used to carry out this study.

These sensitivities are used mainly to interpret the optimization results. It can be seen from these figures that the original design is not at all sensitive to the diameter of the stiffening guide rails (x_2). The stiffening guides are intended to supply sufficient resistance to the out of plane deformation. The results of the sensitivity analyses presented here show that these components fail in this role.

Conversely, it may be noted that in the case of the improved design, the displacement function, calculated via FEM, is sensitive to x_2 and x_3, representing the plates connecting the adjacent stiffening guides. This demonstrates that the components required to prevent unwanted deformation are successful in increasing the stiffness of the assembly.



(a) Sensitivity to x_1.

(b) Sensitivity to x_2.

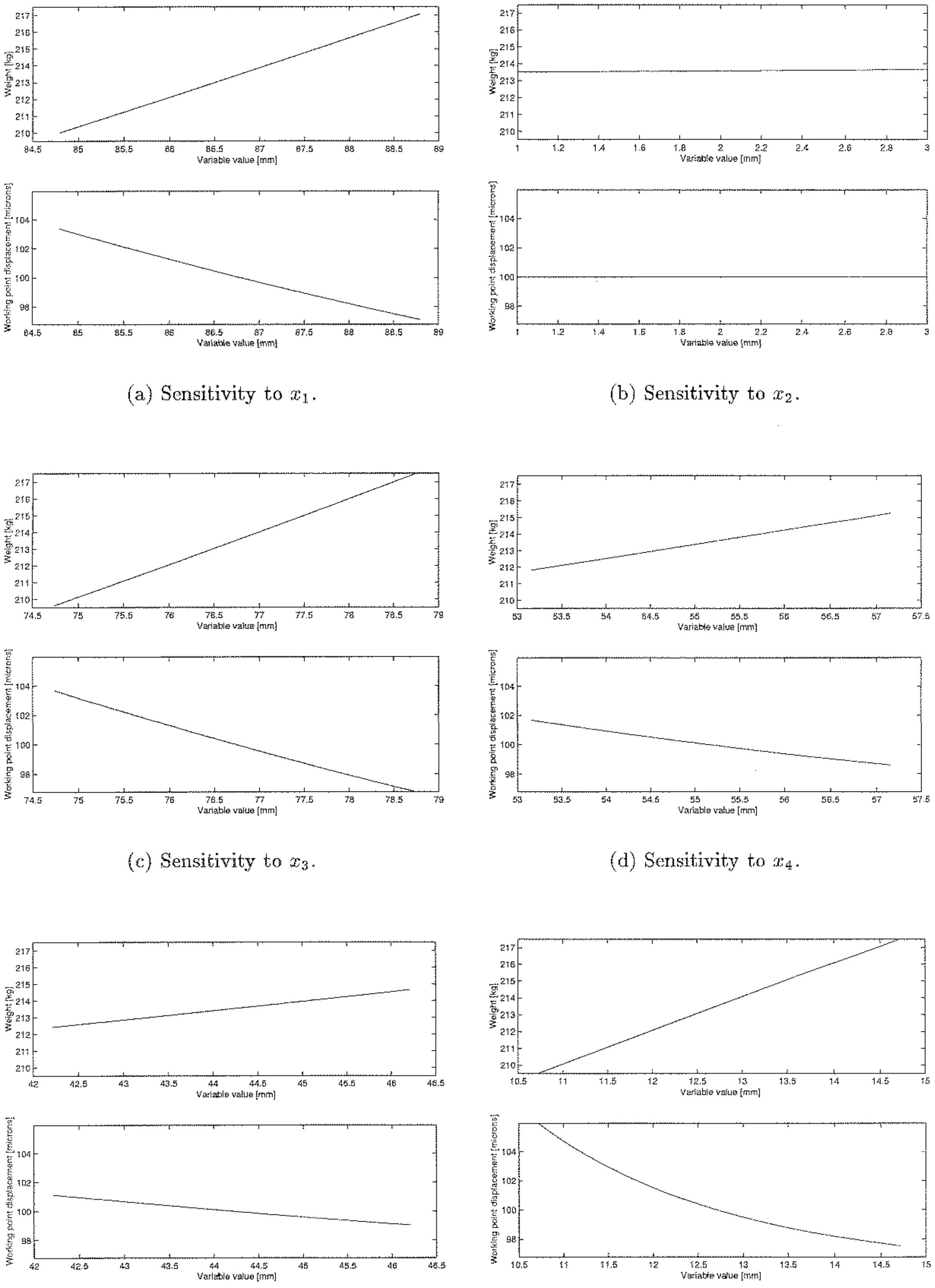
(c) Sensitivity to x_3.

(d) Sensitivity to x_4.

(e) Sensitivity to x_5.

(f) Sensitivity to x_6.

Figure H.1: Function sensitivity in vicinity of terminal variable values (Original design with constraint set $\boldsymbol{g}(\boldsymbol{x})$) and $\bar{d} = 1000\mu\text{m}$).



(a) Sensitivity to x_1.

(b) Sensitivity to x_2.

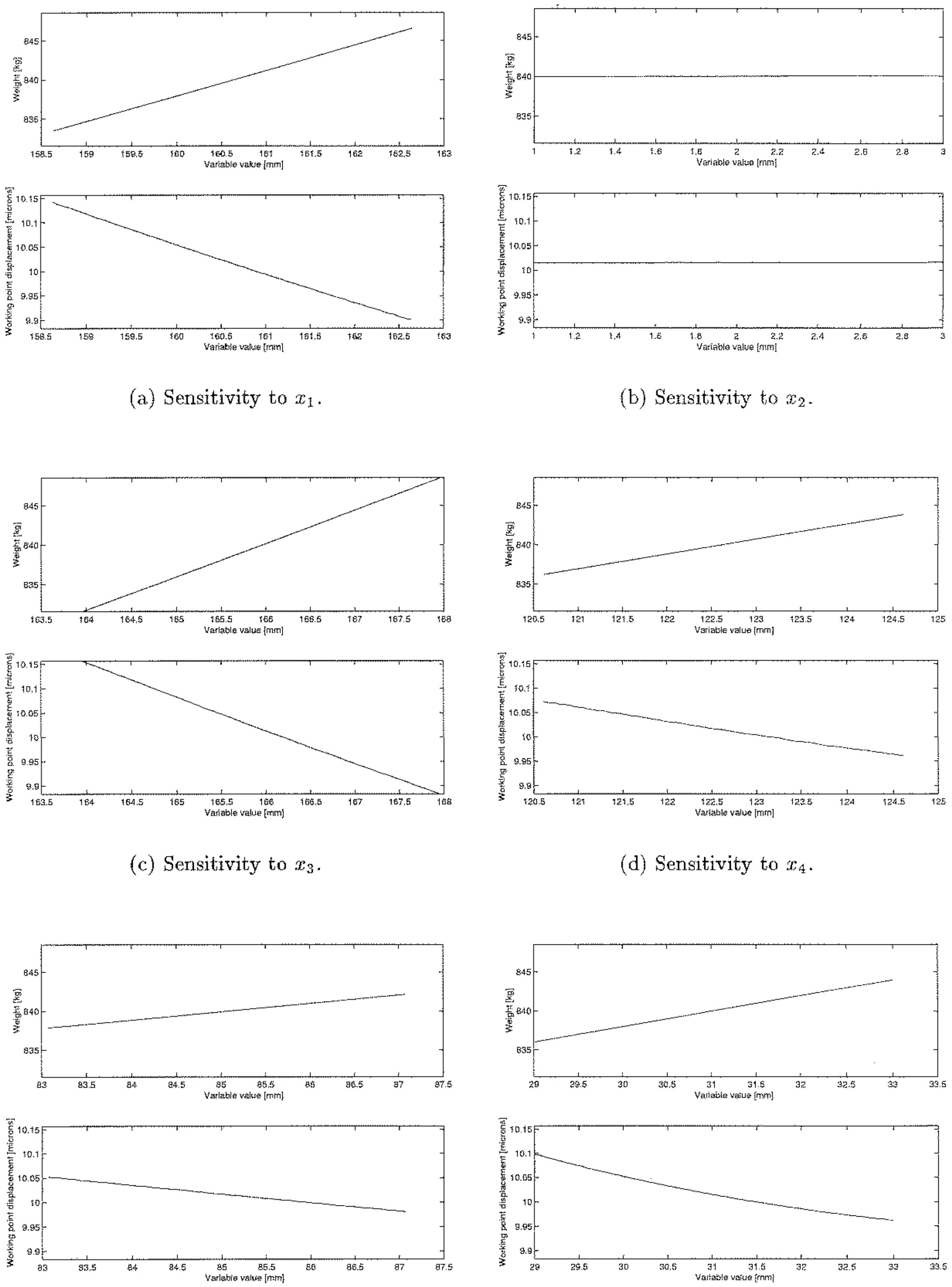
(c) Sensitivity to x_3.

(d) Sensitivity to x_4.

(e) Sensitivity to x_5.

(f) Sensitivity to x_6.

Figure H.2: Function sensitivity in vicinity of terminal variable values (Original design with constraint set $\boldsymbol{g}(\boldsymbol{x})$) and $\bar{d} = 100\mu\text{m}$).



(a) Sensitivity to x_1.

(b) Sensitivity to x_2.

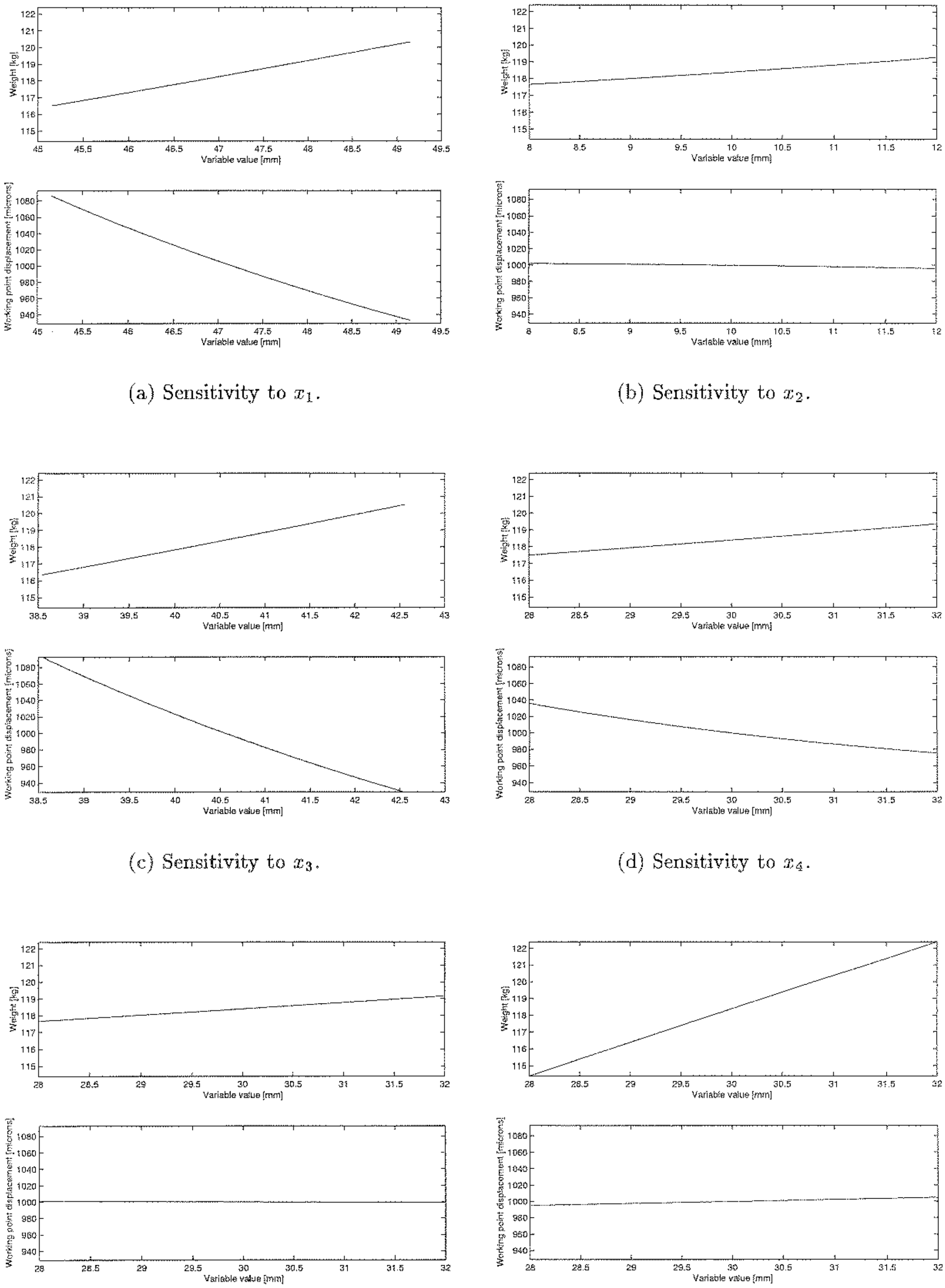
(c) Sensitivity to x_3.

(d) Sensitivity to x_4.

(e) Sensitivity to x_5.

(f) Sensitivity to x_6.

Figure H.3: Function sensitivity in vicinity of terminal variable values (Original design with constraint set $\boldsymbol{g}(\boldsymbol{x})$) and $\bar{d} = 10\mu\text{m}$).



(a) Sensitivity to x_1.

(b) Sensitivity to x_2.

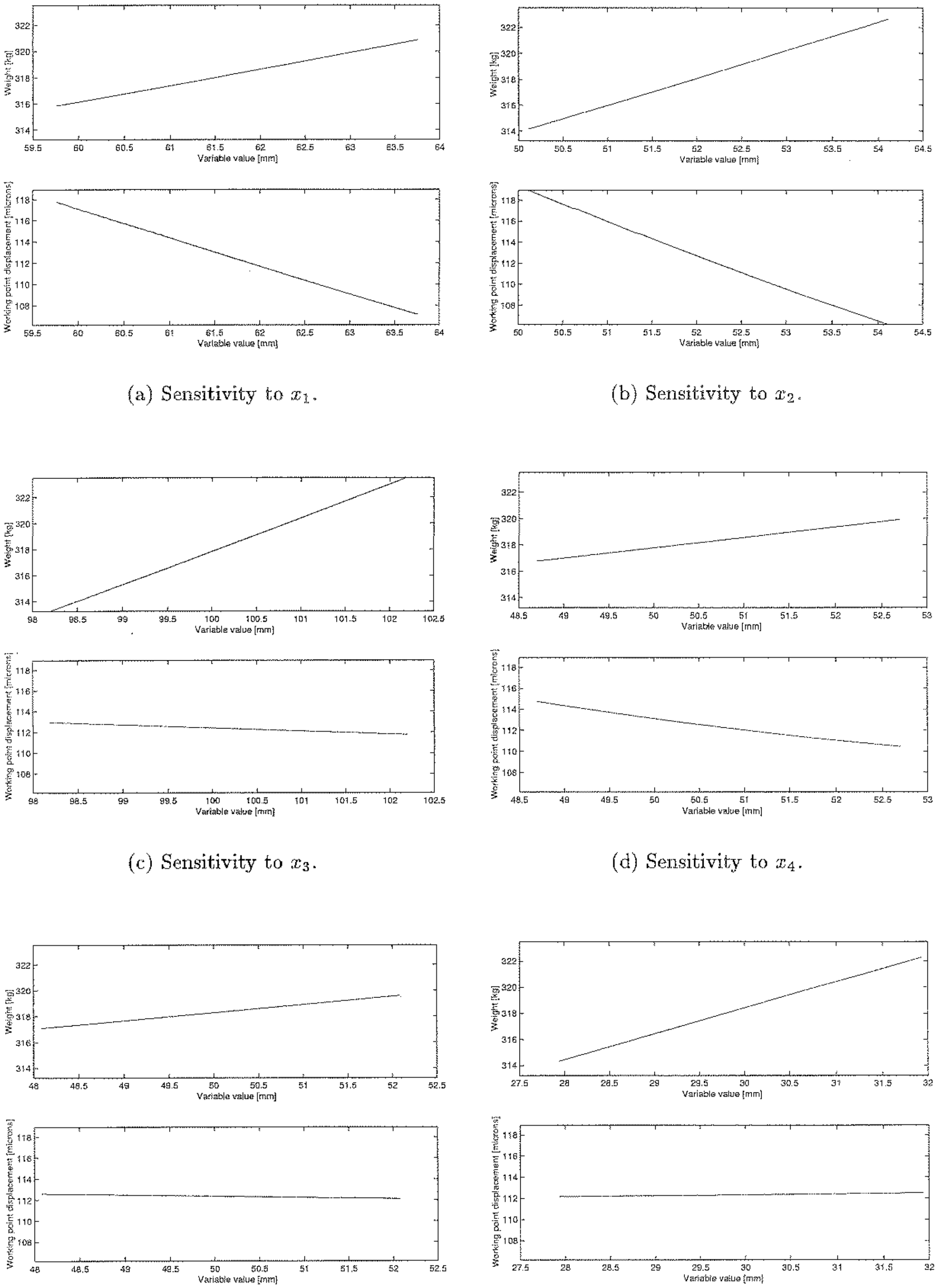
(c) Sensitivity to x_3.

(d) Sensitivity to x_4.

(e) Sensitivity to x_5.

(f) Sensitivity to x_6.

Figure H.4: Function sensitivity in vicinity of terminal variable values (Original design with constraint set $\boldsymbol{G}(\boldsymbol{x})$) and $\bar{d} = 1000\mu$m).



(a) Sensitivity to x_1.

(b) Sensitivity to x_2.

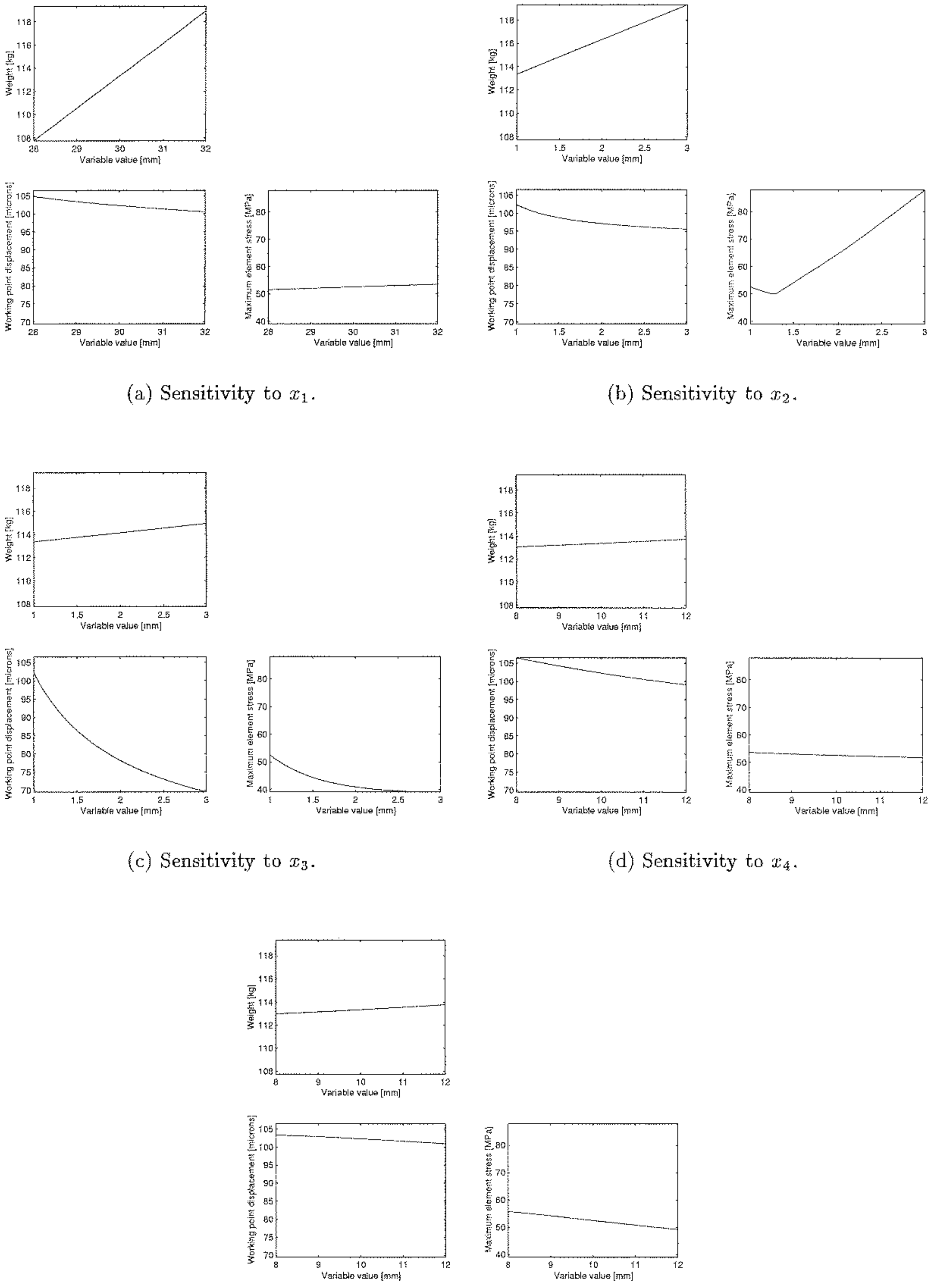
(c) Sensitivity to x_3.

(d) Sensitivity to x_4.

(e) Sensitivity to x_5.

(f) Sensitivity to x_6.

Figure H.5: Function sensitivity in vicinity of terminal variable values (Original design with constraint set $\boldsymbol{G}(\boldsymbol{x})$) and $\bar{d} = 100\mu\text{m}$).



(a) Sensitivity to x_1.

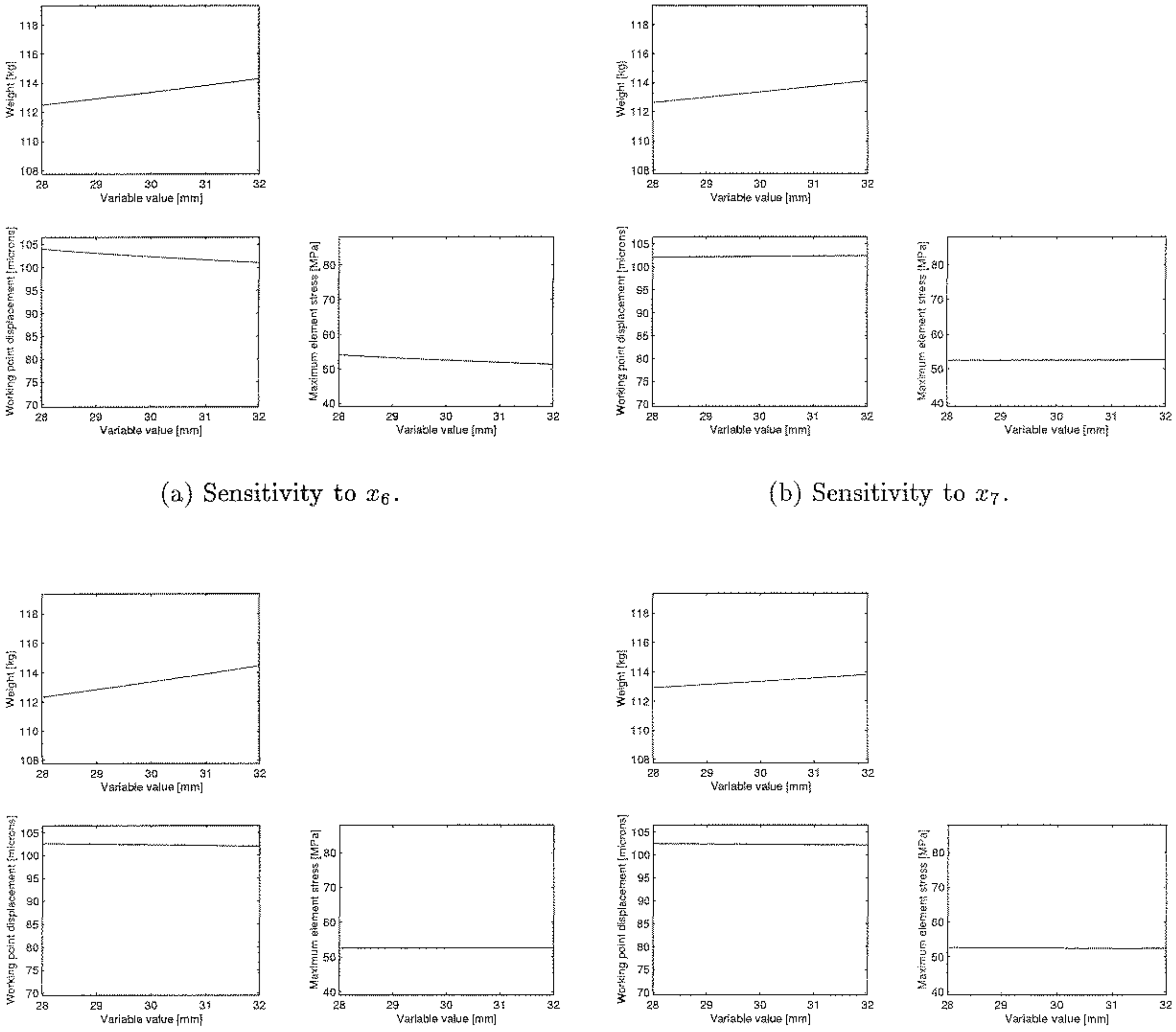
(b) Sensitivity to x_2.

(c) Sensitivity to x_3.

(d) Sensitivity to x_4.

(e) Sensitivity to x_5.

Figure H.6: Function sensitivity in vicinity of terminal variable values (Improved design with $\bar{d} = 1000\mu$m and x_1 to x_5).



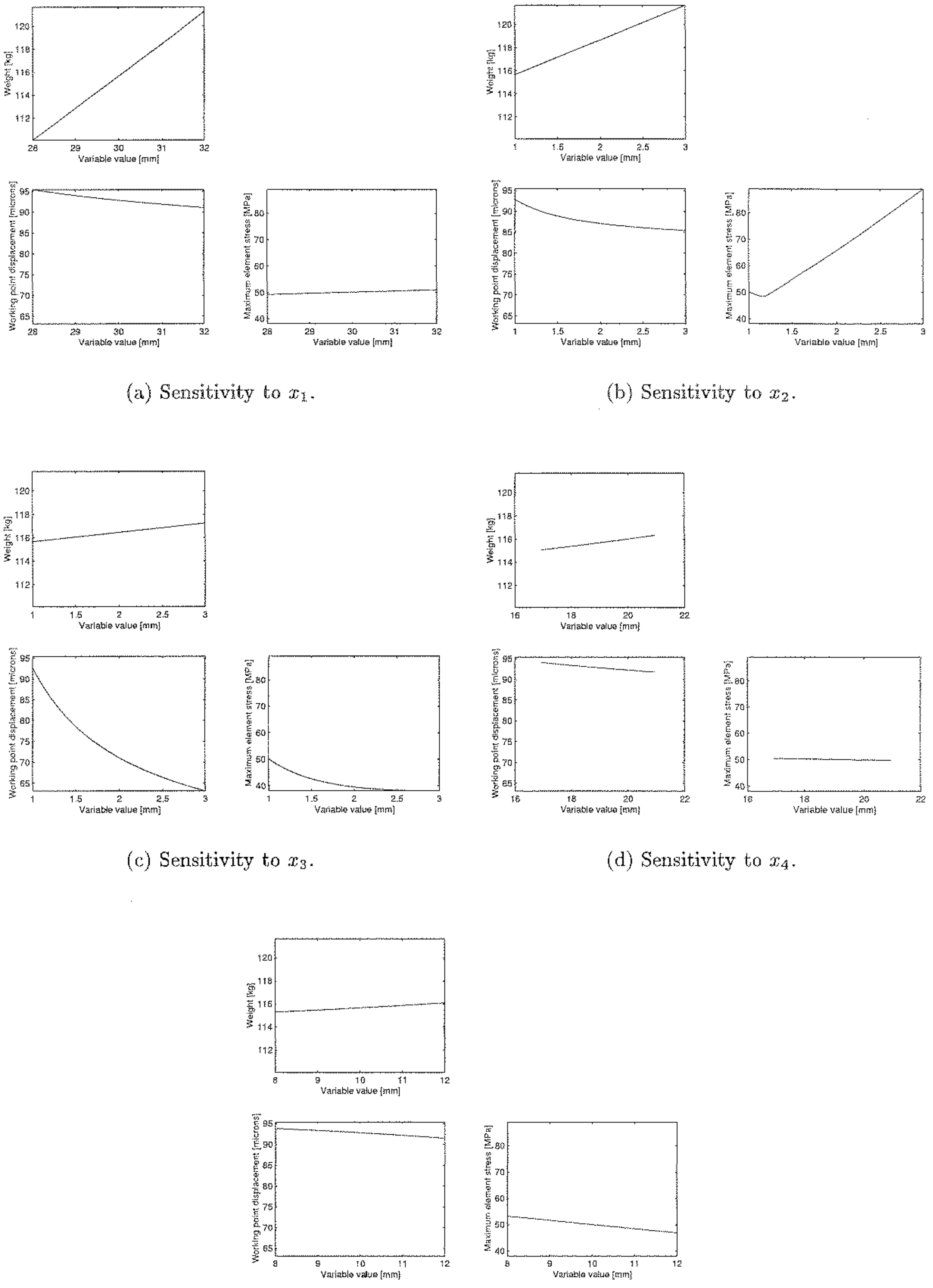
(a) Sensitivity to x_6.

(b) Sensitivity to x_7.

(c) Sensitivity to x_8.

(d) Sensitivity to x_9.

Figure H.7: Function sensitivity in vicinity of terminal variable values (Improved design with $\bar{d} = 1000\mu$m and x_6 to x_9).



(a) Sensitivity to x_1.

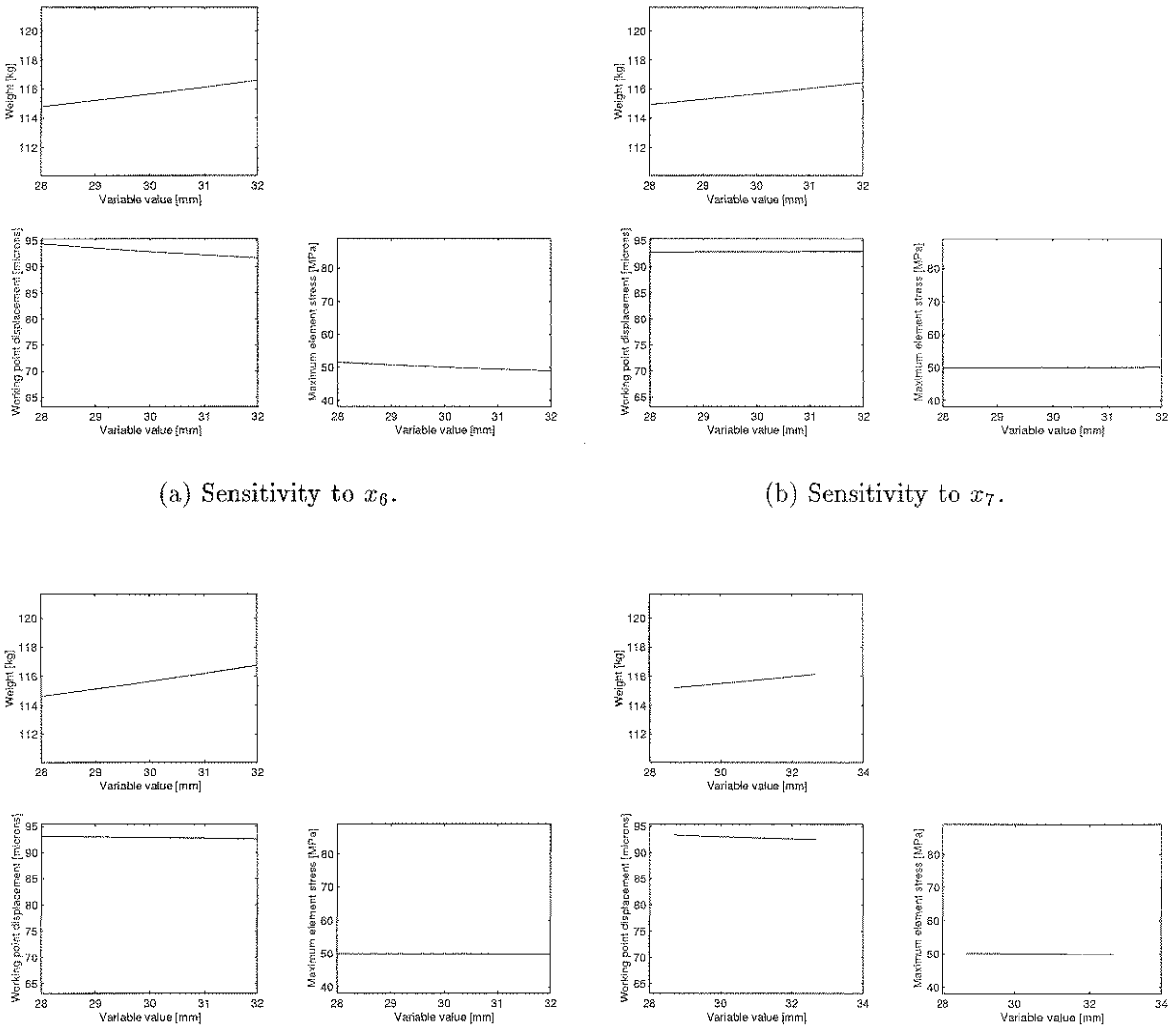
(b) Sensitivity to x_2.

(c) Sensitivity to x_3.

(d) Sensitivity to x_4.

(e) Sensitivity to x_5.

Figure H.8: Function sensitivity in vicinity of terminal variable values (Improved design with $\bar{d} = 100\mu$m and x_1 to x_5).



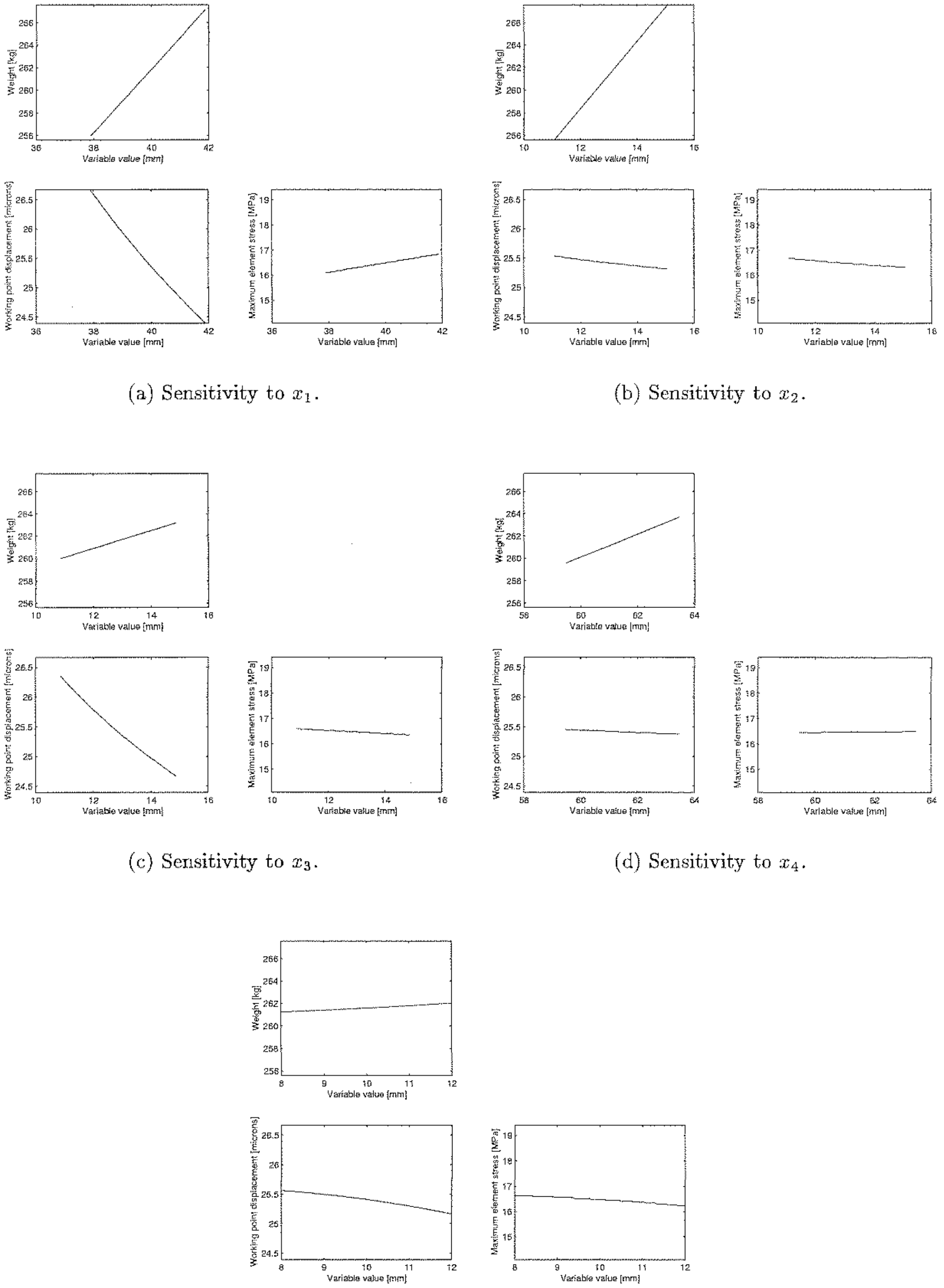
(a) Sensitivity to x_6.

(b) Sensitivity to x_7.

(c) Sensitivity to x_8.

(d) Sensitivity to x_9.

Figure H.9: Function sensitivity in vicinity of terminal variable values (Improved design with $\bar{d} = 100\mu$m and x_6 to x_9).



(a) Sensitivity to x_1.

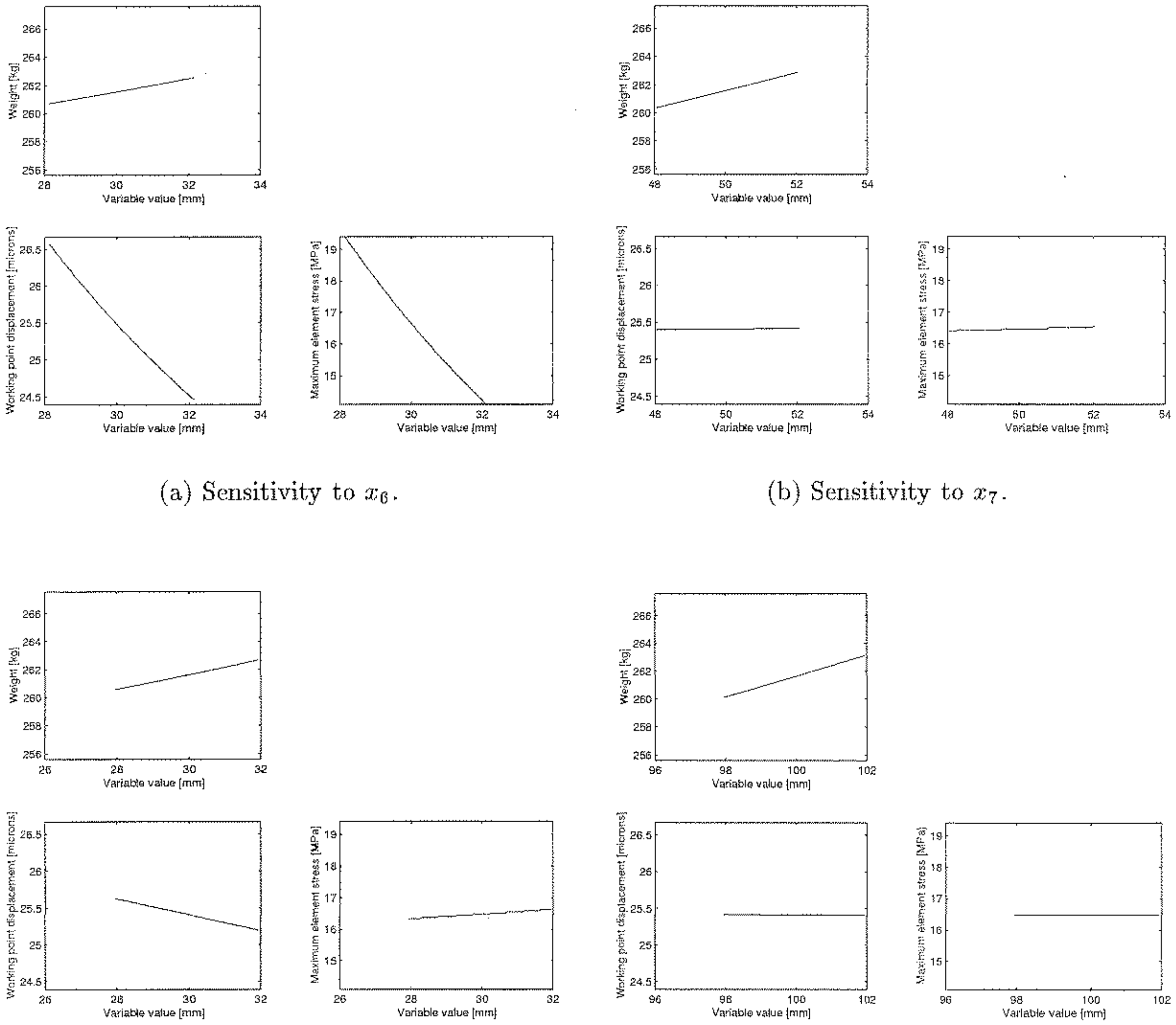
(b) Sensitivity to x_2.

(c) Sensitivity to x_3.

(d) Sensitivity to x_4.

(e) Sensitivity to x_5.

Figure H.10: Function sensitivity in vicinity of terminal variable values (Improved design with $\bar{d} = 10\mu$m and x_1 to x_5).



(a) Sensitivity to x_6.

(b) Sensitivity to x_7.

(c) Sensitivity to x_8.

(d) Sensitivity to x_9.

Figure H.11: Function sensitivity in vicinity of terminal variable values (Improved design with $\bar{d} = 10\mu$m and x_6 to x_9).